

ITS
Institut
Teknologi
Sepuluh Nopember

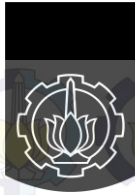
TUGAS AKHIR - TE 145561

RANCANG BANGUN SISTEM OTOMASI PEMBERIAN NUTRISI DAN PENCAHAYAAN UNTUK TAHAP PENYEMAIAN BENIH SELADA PADA PERKEBUNAN SURABAYA HIDROPONIK

Tommy Dwi Putranto
NRP 2213030012
Bayu Fatchur Rohman
NRP 2213030019

Dosen Pembimbing
Suwito, S.T., M.T.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

DESIGN AUTOMATION SYSTEM PROVIDING NUTRITION AND STAGE LIGHTING FOR SEEDING SEEDS LACTUCA SATIVA IN SURABAYA HYDROPONICS GARDEN

Tommy Dwi Putranto
NRP 2213030012
Bayu Fatchur Rohman
NRP 2213030019

Advisor
Suwito, S.T., M.T.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016



PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Rancang Bangun Sistem Otomasi Pemberian Nutrisi dan Pencahayaannya untuk Tahap Penyemaian Benih Selada pada Pekebunan Surabaya Hidroponik”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Tommy Dwi Putranto
NRP 2213030012

Surabaya, 1 juni 2016



Bayu Fatchur Rohman
NRP 2213030019



**RANCANG BANGUN SISTEM OTOMASI PEMBERIAN
NUTRISI DAN PENCAHAYAAN UNTUK TAHAP
PENYEMAIAN BENIH SELADA PADA PERKEBUNAN
SURABAYA HIDROPONIK**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya**

Pada

**Bidang Studi Komputer Kontrol
Program Studi D3 teknik Elektro
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing



**SURABAYA
JUNI, 2016**



RANCANG BANGUN SISTEM OTOMASI PEMBERIAN NUTRISI DAN PENCAHAYAAN UNTUK TAHAP PENYEMAIAN BENIH SELADA PADA PERKEBUNAN SURABAYA HIDROPONIK

Nama : Tommy Dwi Putranto
Bayu Fatchur Rohman
Pembimbing : Suwito, S.T., M.T.

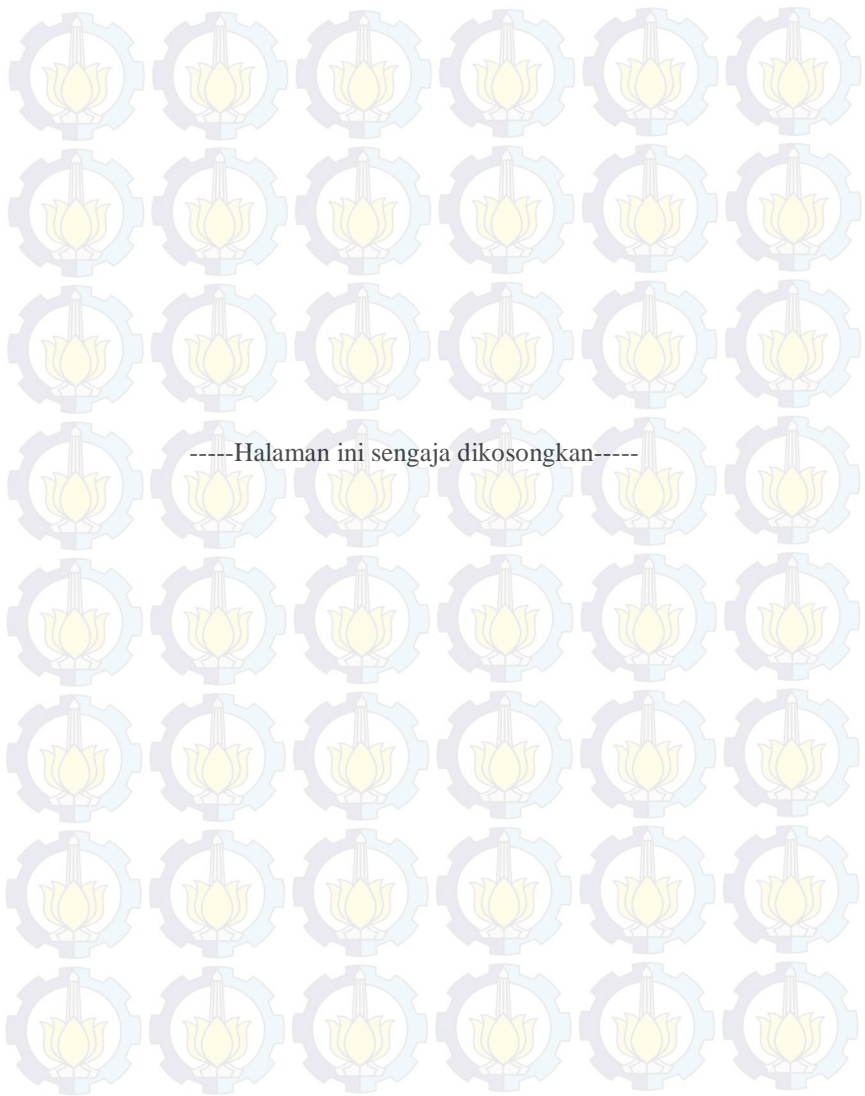
ABSTRAK

Salah satu proses yang sangat menentukan hasil penamaan hidropnik adalah tahap menyemai benih. Pada tahap ini benih tanaman memerlukan penanganan khusus dimana intensitas cahaya matahari tidak boleh terlalu tinggi dan terlalu rendah serta nutrisi harus dijaga agar sesuai dengan kebutuhan EC (*Electrical Conductivity*) yaitu berkisar antara 900 – 1200 uS/cm (*mikro Siemens per centimeter*).

Cara kerja dari alat yang telah dibuat yaitu pada saat hari pertama sampai hari ke tujuh menyalakan pompa air setiap pukul 07.00 – 07.15 WIB dan pukul 17.00 – 17.15 WIB. Lalu pada hari ke 8 sampai hari ke 14 pompa nutrisi AB mix dan sensor EC (*Electrical Conductivity*) baru aktif untuk medeteksi kadar EC pada air dan pompa air akan terus mengalir. Serta mendeteksi EC pada air yang mengalir dengan beberapa aturan yaitu bila kadar EC kurang dari 900 uS/cm , pompa nutrisi aktif. Jika kadar EC antara 900 – 1200 uS/cm pompa nutrisi mati. Jika kadar EC pada air lebih dari 1200 uS/cm alarm akan aktif. Lalu sensor LDR akan aktif mendeteksi cahaya mulai hari pertama sampai hari terakhir yaitu hari ke 14 dari pukul 06.00 – 18.00 WIB.

Hasil dari sistem ini yaitu didapatkan nilai rata – rata kesalahan dari pengukuran EC sebesar 0.1678 % dengan pemberian nutrisi pada hari ke 8 sampai 14 saat selada memiliki 3 atau 4 helai daun.

Kata Kunci : EC (*Electrical Conductivity*), Hidropnik, intensitas, otomasi



DESIGN AUTOMATION SYTEM PROVIDING NUTRITION AND STAGE LIGHTING FOR SEEDING SEEDS LACTUCA SATIVA IN SURABAYA HYDROPONICS GARDER

Name : Tommy Dwi Putranto
Bayu Fatchur Rohman
Advisor : Suwito, S.T., M.T.

ABSTRACT

Our process that will determine the result of hydroponic cultivation is to sow the seed stage. At this stage of seed plants require special handling in which the light intensity should not be too high and should also not be too low as well as nutrients should be kept in accordance with the needs of the EC (Electrical Conductivity) ranged between 900-1200 uS/cm (mikro Siemens per centimeter).

With this situation we find the concept of design automation system providing nutrition and lighting for seeding stage plant hydroponic Lactuca sativa in plantation Surabaya. The works of this system is to make an automated system controlling the concentration of nutrients and intensity of light received by the plant. In this system we use EC Sensor for controlling the nutrients and LDR for controlling light. State specific nutrients and light will transmit the received voltage microcontroller. If the handling of the nutritional of content is less than 900 uS/cm of nutrients will be lit. If nutrition of content is more than 1200 uS/cm alarm on. If plants lack of light, microcontroller will active growlight handling with particular intensity. With such a system is expected to produce seeds hydroponic crops are good and ready to be cultivated.

The result of system measurement Eletrical Conductivity is it have error 0.1678 % and 3 leaf.

Keywords : EC (Electrical Conductivity), Hydroponics, intensity, automation.



KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Program Studi D3 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN SISTEM OTOMASI PEMBERIAN NUTRISI DAN PENCAHAYAAN UNTUK TAHAP PENYEMAIAN BENIH SELADA PADA PERKEBUNAN SURABAYA HIDROPONIK

Dalam Tugas Akhir ini dirancang suatu sistem otomasi pengendalian kepekatan nutrisi dan intensitas cahaya yang diterima oleh tanaman pada perkebunan Hidroponik di Surabaya.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Suwito, S.T., M.T. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 1 juni 2016

Penulis



DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	1
1.3 Batasan Masalah.....	1
1.4 Tujuan	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan	2
1.7 Relevansi.....	3
BAB II TEORI DASAR.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Hidroponik	6
2.2.1 Faktor Mempengaruhi Pertumbuhan Hidroponik.....	7
2.2.2 Cara Pemberian Nutrisi	7
2.2.3 Cara Menaikan PPM Larutan Nutrisi	8
2.2.4 Selada	8
2.2.5 <i>Rockwool</i>	9
2.2.6 Bak Reservoir	9
2.2.7 <i>Nutrient Film Technique</i> (NFT)	10
2.3 Arduino Uno.....	11
2.4 Modul SIM 900	12
2.5 Pompa Air	12
2.6 LED <i>Growlight</i>	12

2.7	Relay.....	13
2.8	<i>Power Supply</i>	13
2.9	Transformator 5A.....	14
2.10	<i>Voltage Regulator</i> LM7805, LM7809, dan LM7812.....	14
2.11	Kapasitor.....	15
2.12	Dioda <i>Full – Wave Rectifier (Diode Bridge)</i>	15
2.13	<i>Real Time Clock (RTC)</i>	16
2.14	Sensor LDR (<i>Light Dependent Resistors</i>)	16
2.15	Sistem Operasi Android	17
2.16	LCD QC1602A	18
2.17	Arduino IDE	19
2.18	Android Studio	19
2.19	Sensor YL – 69	20
2.20	<i>Web Hosting</i>	20
2.21	<i>Database Server</i>	21
2.22	JSON	21
2.23	API (<i>Application Program Interface</i>)	23
BAB III PERANCANGAN SISTEM KONTROL		25
3.1	Perancangan <i>Hardware</i> (Perangkat Keras)	25
3.2	Perancangan Kontroler Arduino Uno.....	27
3.3	Perancangan Power Supply	29
3.4	Perancangan Rangkaian Pengondisi Sinyal Sensor EC	30
3.5	Perancangan Rangkaian Pengondisi Sinyal LDR	32
3.6	Perancangan RTC (<i>Real Time Clock</i>)	33
3.7	Perancangan <i>Driver Relay</i>	33
3.8	Perancangan LED <i>Growlight</i>	35
3.9	Perancangan Rangkaian Pengatur Intensitas <i>Growlight</i>	36
3.10	Perancangan Rangkaian Indikator	37
3.11	Perancangan Mekanik Pompa Nutrisi	38
3.12	Perancangan <i>Hardware</i> Panel Kontrol.....	38
3.13	Perancangan <i>Hardware Plant</i>	40
3.14	Perancangan <i>Web Hosting</i>	41
3.15	Perancangan Database	41
3.16	Implementasi API dan JSON <i>Parsing</i>	46
3.17	Perancangan Aplikasi <i>Mobile</i>	48
BAB IV PENGUKURAN DAN ANALISA		51
4.1	Cara Kerja Alat	51
4.2	Pengujian <i>Hardware</i> (Perangkat Keras).....	51

4.2.1	Pengujian <i>Power Supply</i>	51
4.2.2	Pengujian Rangkaian Driver <i>Relay</i>	53
4.2.3	Pengujian Lampu LED <i>Growlight</i>	54
4.2.4	Pengujian Pompa Nutrisi.....	54
4.2.5	Pengujian Sensor EC (<i>Electrical Conductivity</i>)	55
4.2.6	Pengujian Modul SIM 900	55
4.2.7	Pengujian RTC (<i>Real Time Clock</i>)	56
4.3	Pengujian <i>Software</i> (Perangkat Lunak)	57
4.4	Pengujian Keseluruhan	57
4.4.1	Hari Kurang Dari Tujuh.....	58
4.4.2	Hari Lebih Dari Tujuh.....	61
BAB V PENUTUP		67
DAFTAR PUSTAKA		69
LAMPIRAN A		71
A.1.	Program Arduino	71
A.2.	Program Android Studio	81
A.3.	Program <i>Update Server</i>	84
A.4.	Program <i>JSON</i>	85
DAFTAR RIWAYAT HIDUP		87
DAFTAR RIWAYAT HIDUP		89



DAFTAR GAMBAR

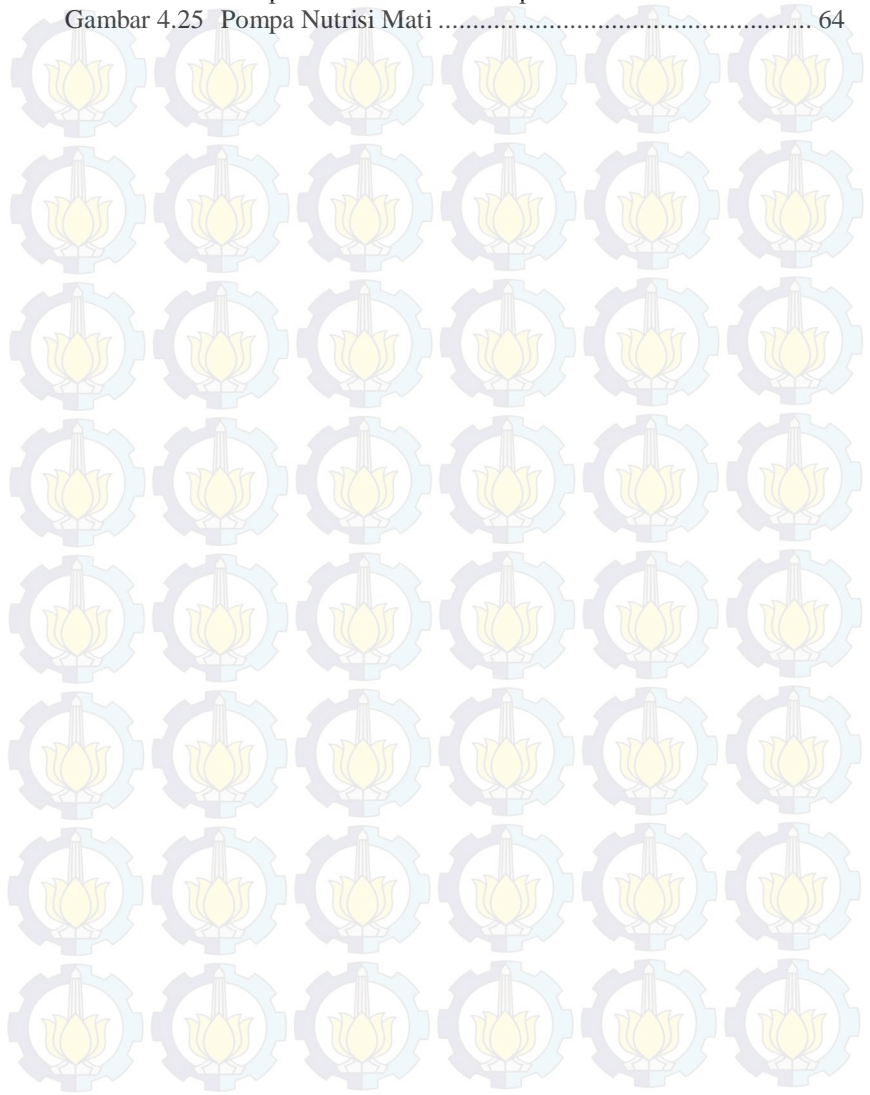
	HALAMAN
Gambar 2.1 Nutrisi AB – MIX.....	7
Gambar 2.2 Bak Reservoir.....	10
Gambar 2.3 Sistem NFT	10
Gambar 2.4 Arduino Uno	11
Gambar 2.5 GPRS Shield SIM 900.....	11
Gambar 2.6 Pompa Air	12
Gambar 2.7 LED <i>Growlight</i>	13
Gambar 2.8 Rangkaian <i>Power Supply</i>	14
Gambar 2.9 Rangkaian <i>Diode Bridge</i>	15
Gambar 2.10 Rangkaian <i>Real Time Clock</i>	16
Gambar 2.11 Arsitektur Sistem Operasi Android	17
Gambar 2.12 Modul LCD QC1602A	18
Gambar 2.13 Tampilan Ardunio IDE	18
Gambar 2.14 Tampilan <i>Software</i> Android Studio	19
Gambar 2.15 Gambaran Umum <i>Web Hosting</i>	21
Gambar 2.16 Contoh Data JSON	22
Gambar 2.17 Arsitektur <i>JSON Parsing</i>	23
Gambar 3.1 Skema Prinsip Kerja Sistem.....	26
Gambar 3.2 Skematik <i>Shield</i> Tambahan.....	27
Gambar 3.3 Skematik Rangkaian <i>Power Supply</i>	30
Gambar 3.4 Skematik Rangkaian Pengondisi Sinyal Sensor EC.....	31
Gambar 3.5 Skematik Rangkaian Pengondisi Sinyal LDR	32
Gambar 3.6 Skematik Rangkaian RTC	32
Gambar 3.7 Rangkaian <i>Driver Relay</i>	33
Gambar 3.8 Panjang Gelombang Spektrum Warna LED	34
Gambar 3.9 Skematik Rangkaian <i>Growlight</i>	35
Gambar 3.10 Perancangan Mekanik LED <i>Growlight</i>	36
Gambar 3.11 Skematik Rangkaian Pengatur Intensitas <i>Growlight</i>	36
Gambar 3.12 Skematik Rangkaian Indikator.....	37
Gambar 3.13 Skematik Rangkaian Level Nutrisi.....	37
Gambar 3.14 Perancangan Mekanik Pompa Nutrisi	38
Gambar 3.15 Perancangan Panel Bagian Dalam.....	39
Gambar 3.16 Tampilan Panel.....	39

Gambar 3.17	Desain Prototipe Hidroponik Tampak Atas	40
Gambar 3.18	Desain Prototipe Hidroponik Tampak Depan	41
Gambar 3.19	Tampilan CPanel Secara <i>Cloud</i>	42
Gambar 3.20	Halaman <i>MySQL Database Wizard</i>	42
Gambar 3.21	Tampilan Membuat Database <i>Users</i>	43
Gambar 3.22	Tampilan Step Tiga Dalam Membuat Database	43
Gambar 3.23	Tampilan <i>Php My Admin</i>	44
Gambar 3.24	Struktur Tabel	44
Gambar 3.25	<i>Source Code</i> Php Untuk Mengkoneksikan ke Database	45
Gambar 3.26	Contoh Tampilan Url	45
Gambar 3.27	Tampilan Url	45
Gambar 3.28	<i>Source Code</i> API	46
Gambar 3.29	Tampilan <i>JSON Code</i>	47
Gambar 3.30	Arsitektur Aplikasi	47
Gambar 3.31	Tampilan Aplikasi Android	48
Gambar 3.32	Skema Perangkat Lunak	49
Gambar 3.33	Diagram Alur Program	49
Gambar 4.1	<i>Power Supply</i>	52
Gambar 4.2	<i>Pengujian Driver Relay</i>	53
Gambar 4.3	<i>Pengujian Growlight</i>	53
Gambar 4.4	<i>Pengujian Pompa Nutrisi</i>	54
Gambar 4.5	Url Menampilkan Data Oleh Modul SIM 900	55
Gambar 4.6	Tampilan <i>JSON</i>	56
Gambar 4.7	<i>Pengujian RTC</i>	56
Gambar 4.8	<i>Pengujian Tampilan Aplikasi Android</i>	57
Gambar 4.9	Pompa Air Belum Aktif	58
Gambar 4.10	Waktu Pada Sistem	58
Gambar 4.11	Penunjuk Waktu	59
Gambar 4.12	Pompa Air Aktif	59
Gambar 4.13	Waktu Menunjukkan Pukul 17.17	59
Gambar 4.14	Pompa Air Berhenti	60
Gambar 4.15	Waktu Menunjukkan Pukul 07.02	60
Gambar 4.16	Pompa Air Aktif	60
Gambar 4.17	Waktu Menunjukkan Pukul 07.16	61
Gambar 4.18	Pompa Air Berhenti	61
Gambar 4.19	Pengaduk	62
Gambar 4.20	Pompa Air Aktif	62
Gambar 4.21	Hari Menunjukkan Lebih Dari Tujuh	62
Gambar 4.22	Tampilan Kadar EC Pada Aplikasi Android	63

Gambar 4.23 Pompa Nutrisi Aktif 63

Gambar 4.24 Tampilan Kadar EC Pada Aplikasi Android..... 64

Gambar 4.25 Pompa Nutrisi Mati 64





DAFTAR TABEL

	HALAMAN
Tabel 2.1	Tabel Pemberian Nurisi AB – MIX Hidroponik..... 6
Tabel 3.1	Konfigurasi Port <i>Shield</i> Tambahan 28
Tabel 4.1	Hasil Pengukuran <i>Power Supply</i> 52
Tabel 4.2	Tabel Pengukuran Nilai EC 55



BAB I

PENDAHULUAN

1.1 Latar Belakang

Tahap – tahap dalam budidaya hidroponik dibagi menjadi 3 yaitu tahap pembibitan, tahap remaja, dan tahap dewasa. Tahap pembibitan merupakan tahap yang paling rawan. Karena jika tahap ini gagal maka tahap – tahap berikutnya tidak dapat dilakukan. Sehingga permintaan sayur oleh pelanggan tertunda bahkan tidak dapat terpenuhi. Tahap pembibitan memerlukan waktu 2 minggu. Pada tahap ini diperlukan penanganan khusus dimana intensitas cahaya matahari tidak boleh terlalu tinggi dan juga tidak boleh terlalu rendah. Karena apabila pencahayaan terlalu tinggi atau terlalu rendah menyebabkan bibit yang dihasilkan tidak optimal yaitu mengalami etiolasi. Etiolasi adalah keadaan dimana pertumbuhan tumbuhan yang sangat cepat namun kondisi tumbuhan lemah, batang tidak kokoh, daun kecil, dan tumbuhan tampak pucat.

Selain itu faktor yang mempengaruhi proses pembibitan adalah kadar nutrisi harus di jaga agar tetap stabil. Perlakuan ini perlu dilakukan karena jika kadar nutrisi terlalu rendah maka pertumbuhan dari benih menjadi bibit akan lambat. Satuan untuk menyatakan kadar nutrisi dalam air adalah ppm (part per million) atau $\mu\text{S/cm}$ (*micro Siemens per centimeter*). Kadar nutrisi yang diperlukan untuk proses pembibitan berkisar 900 – 1200 $\mu\text{S/cm}$.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah menjaga nutrisi yang diterima oleh tanaman tetap pada 900 – 1200 $\mu\text{S/cm}$ serta menjaga intensitas cahaya yang diterima tanaman tetap ada pada lingkungan kurang mendukung misalnya saat cuaca berawan.

1.3 Batasan Masalah

Faktor – faktor yang mempengaruhi sistem hidroponik yaitu suhu dan pH yang terkandung dalam bak nutrisi dianggap dalam keadaan ideal. Serta keadaan komponen LDR yang di gunakan untuk sensor cahaya berada dalam keadaan ideal.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang suatu sistem otomasi pemberian nutrisi untuk tanaman hidroponik pada tahap pembibitan agar tetap pada kadar yang stabil dan menjaga intensitas cahaya yang diterima oleh tanaman meskipun dalam kondisi cuaca yang berawan. Hasil yang diharapkan adalah dapat menghasilkan bibit tanaman hidroponik yang baik dan siap dibudidayakan.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perancangan sistem, perancangan alat, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai identifikasi fisik tentang hidroponik, studi tentang arduino dan karakteristik baik secara *hardware* dan *software*, mempelajari EC (*Electrical Conductivity*), mengidentifikasi modul sim 900, studi tentang pengkondisian sistem hidroponik NFT, mempelajari *cloud server*, mempelajari mengintegrasikan modul sim 900 dengan arduino, konsep growlight, konsep pemrograman OS android, serta solusi untuk memperoleh nilai EC berkisar antara 900 – 1200 uS/cm. Pada tahap perancangan sistem dan perancangan alat, sensor EC yang telah dibuat dikalibrasi agar dapat dibaca oleh arduino. Lalu data tersebut di unggah ke *cloud server* oleh modul sim 900. Setelah itu data yang telah berada di *cloud* ditampilkan dalam perangkat OS Android. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, sistem hidroponik NFT, konsep EC (*Electrical Conductivity*), karakteristik arduino, karakteristik modul sim900, konsep *cloud server*, konsep *growlight*, pemrograman OS Android, serta memperoleh nilai EC yang cocok untuk tanaman hidroponik.

Bab III Perancangan Sistem

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain mekanik dan perangkat lunak (*Software*) yang meliputi program yang akan digunakan untuk menjalankan alat tersebut

Bab IV Pengukuran dan Analisa

Bab ini memuat hasil implementasi serta analisis dari hasil alat tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi alat untuk mendapatkan bibit selada yang siap di budidaya pada perkebunan hidroponik. Selain itu, bisa sebagai salah satu solusi untuk mendapatkan tambahan pendapatan rumah tangga.



BAB II

TEORI DASAR

2.1 Tinjauan Pustaka

Hidroponik adalah suatu sistem bercocok tanam dengan menggunakan air sebagai media tanamnya. Air yang digunakan hendaknya memenuhi syarat – syarat tertentu, misalnya kadar nutrisi, kekeruhan, ukuran partikel, dan proporsi[4].

Pemberian nutrisi hidroponik yang tepat akan memberikan hasil yang optimal bagi pertumbuhan selada. Selain itu pertumbuhan tanaman tidak lepas dari lingkungan tumbuh terutama faktor media tanam yang secara langsung akan mempengaruhi hasil tanaman[1]. Menurut Darmawan (1997) pertumbuhan selada akan optimal pada kisaran suhu udara $25^{\circ} - 26^{\circ} \text{C}$ dan kelembapan berkisar antara 76 – 77 %. Keadaan suhu di dalam ruangan berkisar antara $27,8^{\circ} - 33,9^{\circ} \text{C}$ dengan kelembapan antara 58,17% - 75,5%. Dalam budidaya secara hidroponik, perlu diperhatikan kondisi pH dan EC larutan nutrisi. pH yang baik untuk tanaman selada berkisar antara 6,0 – 6,5 dengan EC antara 1,5 – 2,5 dS/m [2].

Aspek penting yang perlu juga diperhatikan dalam menentukan keberhasilan budidaya hidroponik adalah pengelolaan tanaman yang meliputi persiapan bahan media, larutan nutrisi, pemeliharaan, dan aplikasi larutan nutrisi [3].

Pada tanaman hidroponik, pemberian Nutrisi dengan kadar EC yang berlebihan akan mengakibatkan tanaman tersebut menjadi berjamur akibat panas yang dihisap[5]. Karena pada saat proses pembibitan dari hari 0 sampai hari ke 14 merupakan hari yang sangat vital untuk proses pembibitan tanaman hidroponik.

Berdasarkan permasalahan tersebut maka dalam Tugas Akhir ini dirancang sebuah perangkat yang mampu mengendalikan kadar EC pada sistem hidroponik pada tanaman selada dengan cara mengendalikan pemberian nutrisi yang berisi cairan nutrisi tipe A dan nutrisi tipe B. komponen pengendalian yang digunakan adalah Arduino mega 2560. Tujuan yang ingin dicapai dalam penelitian ini adalah terbuat perangkat yang mampu mengendalikan kadar EC pada tanaman hidroponik menggunakan arduino mega 2560.

2.2 Hidroponik

Dalam kajian Bahasa, hidroponik berasal dari kata *hydro* yang berarti air dan *ponos* yang berarti kerja. Jadi, hidroponik memiliki pengertian secara bebas teknik bercocok tanam dengan menekankan pada pemenuhan kebutuhan nutrisi bagi tanaman, atau dalam pengertian sehari-hari bercocok tanam tanpa tanah. Dari pengertian ini terlihat bahwa munculnya teknik bertanam secara hidroponik diawali oleh semakin tingginya perhatian manusia akan pentingnya kebutuhan pupuk bagi tanaman. Di mana pun tumbuhnya sebuah tanaman akan tetap dapat tumbuh dengan baik apabila nutrisi (unsur hara) yang dibutuhkan selalu tercukupi. Dalam konteks ini fungsi dari tanah adalah untuk penyangga tanaman dan air yang ada merupakan pelarut nutrisi, untuk kemudian bisa diserap tanaman. Pola pikir inilah yang akhirnya melahirkan teknik bertanam dengan hidroponik, di mana yang ditekankan adalah pemenuhan kebutuhan nutrisi.

Tabel 2.1 merupakan beberapa tanaman hidroponik dengan kebutuhan nutrisi selama masa penanaman. Pada kolom PPM max digunakan hanya sebagai acuan dan tidak harus sama persis dengan tabel namun disarankan.

Tabel 2.1 Tabel Pemberian Nurisi AB – MIX Hidroponik

Nama Tanaman	PPM Max	pH	Masa Panen (Hari)
Pakcoy	1050 – 1400	7	40 – 60
Kangkung	1050 – 1400	5.5 – 6.5	28
Sawi	1050 – 1400	5.5 – 6.5	40 – 60
Cabe	1260 – 1540	6.0 – 6.5	40 – 70
Bayam	1260 – 1610	6.0 – 7.0	25
Seledri	1260 – 1680	6.5	120 – 150
Tomat	1400 – 3500	6.0 – 6.5	63
Selada	560 – 840	6.0 – 7.0	65 – 90
Melon	1400 – 1750	5.5 – 6.0	74
Brokoli	1050 – 1400	6.5 – 7.0	75
Terong	1750 – 2450	6.0	60

2.2.1 Faktor Mempengaruhi Pertumbuhan Hidroponik

Pertumbuhan tanaman hidroponik dipengaruhi oleh berbagai macam faktor diantaranya, nutrisi dan kondisi lingkungannya. Nutrisi hidroponik ini adalah pupuk hidroponik lengkap yang mengandung semua unsur hara makro dan mikro yang diperlukan tanaman hidroponik. Pupuk tersebut diformulasi secara khusus sesuai dengan jenis dan fase pertumbuhan tanaman. Nutrisi hidroponik tersedia untuk berbagai jenis tanaman dengan takaran yang berbeda setiap jenis tanamannya. Oleh karena itu, untuk mengendalikan nutrisi sesuai yang dibutuhkan oleh suatu jenis tanaman menggunakan sensor EC. Nutrisi AB Mix pada Gambar 2.1 mempengaruhi dalam memberi nilai EC (*Electrical Conductivity*) jika nilai berlebih maka tanaman akan menjadi panas dan jika terlalu sedikit akan membuat tanaman menjadi pucat.

2.2.2 Cara Pemberian Nutrisi

Cek dahulu ppm air bakunya, Gunakan air baku dengan ppm rendah (di bawah 100 ppm). Buat larutan nutrisi awal dengan perbandingan 1 – 3 – 3 yang artinya setiap 1 liter air baku ditambah 3 ml pekatan A dan 3 ml pekatan B lalu aduk hingga rata. Hasilnya akan didapat larutan nutrisi dengan kepekatan sekitar 500 ppm. Jika kurang pekat tambahkan nutrisi, jika terlalu pekat tambahkan air. Porsi nutrisi A harus selalu sama dengan porsi nutrisi B.



Gambar 2.1 Nutrisi AB – MIX

2.2.3 Cara Menaikan PPM Larutan Nutrisi

Setiap penambahan 1 ml pekatan A dan 1 ml pekatan B dalam setiap liter larutan nutrisi yang sudah ada akan menaikkan ppm sekitar 130 poin. Atau setiap kenaikan 100 ppm perlu tambahan 0.75 ml nutrisi A dan 0.75 ml nutrisi B untuk setiap liter larutan nutrisi yang sudah ada. Contohnya jika sudah punya 5 liter larutan nutrisi dengan kepekatan 500 ppm dan mau menaikannya menjadi 700 ppm maka harus menambahkan (5×1.5) ml pekatan A dan (5×1.5) ml pekatan B. jadi harus menambah 7.5 ml pekatan A dan 7.5 ml pekatan B ke dalam 5 liter larutan tersebut.

2.2.4 Selada

Selada (*Lactuca sativa*) merupakan tanaman daerah beriklim tropis maupun sedang. Jenis tanaman hidroponik paling populer di tanah air itu merupakan tumbuhan asli bagian timur Laut Tengah. Dalam bahasa romawi, *Lac* yang terselip dalam kata *Lactuca* berarti susu. Mengacu pada cairan putih yang keluar ketika batang dibelah. *Sativa* berarti dibudidayakan. Sejarah mencatat, selada telah ditanam pada zaman Mesir Kuno sejak 4500 SM. Biji dan daunnya menjadi komoditas penting sebagai bahan pangan dan penghasil minyak. Selada yang kaya akan vitamin A dan potassium dibudidayakan dalam suhu yang relatif rendah. Itu untuk mencegah pertumbuhan mengarah ke generatif. Karena kini bagian yang dimanfaatkan lebih banyak berupa daun, tumbuhnya bunga atau pembentukan biji dihindari. Suhu tinggi diketahui memicu pertumbuhan bunga. Suhu sangat dingin pun tidak cocok. Memicu pertumbuhan lambat dan merusak daun tertular.

Sejatinya, pertumbuhan selada dewasa mencapai 65 – 130 hari setelah tanam. Namun, jika dipanen setua itu, rasa selada pahit dan tidak laku dijual. Itu sebabnya panen dilakukan ketika tanaman masih relatif muda. Pemberian tanaman hingga dewasa hanya dilakukan untuk tujuan mendapatkan benih. Pengelompokkan selada berdasarkan kelompok kultivarnya yaitu daun longgar (*loose leaf*), daun memanjang (*romaine* atau *cos*), kepala atau kepala renyah (*crisphead*), kepala dengan susunan daun yang lepas atau kepala mentega (*butterhead*) yang lebih lembut daunnya ketimbang kepada renyah, perpaduan antara *crisphead* dan daun (*summercrisp*), batang (*stem*), dan minyak (*oilseed*). Dari tujuh kelompok itu, hanya tiga yang paling banyak dibudidayakan yaitu daun

longgar, kepala renyah, dan *romaine/cos*. Selada keriting lokal termasuk jenis selada daun longgar.

Harga benih selada mulai Rp. 1.000 per plastic berisi ratusan benih. Bibit selada tumbuh lebih baik pada media *perlite* dan vermikulit dengan perbandingan 1:1 dibandingkan dengan penggunaan media *rockwool*. Itu hasil penelitian di *Cornell University*, Amerika Serikat yang melakukan riset pada *butterhead*. Pekebun komersial tetap memilih *rockwool* karena alasan kepraktisan. Bibit dalam *rockwool* mudah dipindahkan tanpa media berceceran dan hasil panen tetap memuaskan. Tingkat Nutrisi yang dibutuhkan berkisar 900 – 1200 uS/cm atau 560 – 840 PPM dengan suhu sejuk sekitar 20°C. Namun, kini tersedia pula varietas yang tahan di daerah lebih panas. Walaupun tahan panas, selada juga tidak tahan hujan lebat. Sehingga pekebun memilih menanamnya pada akhir musim hujan. Namun, sebenarnya penanaman saat musim hujan dapat disiasati dengan naungan atap maupun net yang mencegah tetesan hujan jatuh langsung di permukaan daun.

2.2.5 Rockwool

Rockwool adalah bahan nonorganik yang dibuat dengan cara meniupkan udara atau uap ke dalam batuan yang dilelehkan. Hasilnya berupa sejenis fiber yang berongga – rongga dengan diameter 6 – 10 mikrometer. Sifatnya mampu menahan air dan udara dalam jumlah yang baik untuk mendukung pertumbuhan akar. Harga *rockwool* ukuran 100 cm x 15 cm x 7.5 cm yang cukup untuk membuat 1500 potong media tanam mencapai Rp 70000 – Rp 90000 tergantung kuantitas dan toko yang menjualnya. Satu *rockwool* mampu menyerap air hingga 15 liter. Untuk pemakaiannya, pekebun perlu memotong *rockwool* kecil – kecil sekitar 2 cm x 2 cm x 2 cm. Bongkahan *rockwool* yang rapi membuatnya gampang diatur dibandingkan media yang berupa butiran.

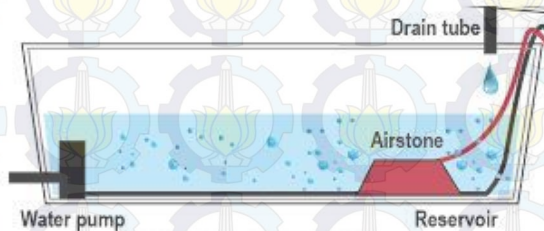
2.2.6 Bak Reservoir

Reservoir pada sistem ini berfungsi sebagai bak pencampuran antara air, nutrisi dan segala hal yang dibutuhkan untuk menjaga kondisi kebutuhan EC (*Electrical Conductivity*) tanaman. Cairan di dalam reservoir ini akan dikondisikan sesuai dengan kebutuhan tanaman hidroponik karena merupakan wadah cairan yang akan diisi (*fill*) ke *plant* hidroponik dan menjadi wadah bagi aliran nutrisi yang tidak diserap tanaman hidroponik untuk diproses kembali (*drain*) sehingga

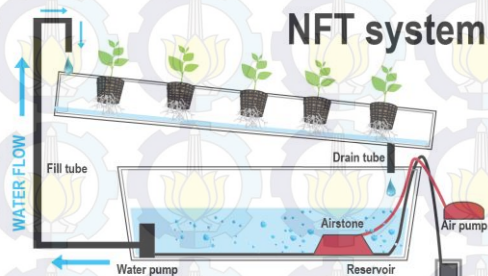
tidak ada cairan nutrisi yang terbuang kecuali diserap tanaman hidroponik sesuai kebutuhannya. Tampilan dari bak reservoir seperti pada Gambar 2.2

2.2.7 Nutrient Film Technique (NFT)

NFT merupakan model budidaya hidroponik dengan meletakkan akar tanaman pada lapisan air yang dangkal. Air tersebut tersirkulasi dan mengandung nutrisi sesuai kebutuhan tanaman. Perakaran bisa berkembang di dalam larutan nutrisi. Karena di sekeliling perakaran terdapat selapis nutrisi, maka sistem ini dikenal dengan nama *nutrient film technique* [9]. Dalam penelitian ini kemiringan pipa talang dalam konstruksi hidroponik NFT yang diterapkan besarnya yaitu 5% setiap meternya. Gambar 2.3 merupakan gambar dari contoh sistem NFT.



Gambar 2.2 Bak Reservoir



Gambar 2.3 Sistem NFT

2.3 Arduino Uno

Arduino adalah perangkat elektronik atau papan rangkaian elektronik *open – source* yang di dalamnya terdapat komponen utama, yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah chip atau IC (*Integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca *input*, memproses *input* tersebut kemudian menghasilkan *output* sesuai yang diinginkan. Jadi, mikrokontroler bertugas sebagai otak yang mengendalikan *input*, proses, dan *output* sebuah rangkaian elektronik. Salah satu tipe Arduino yang akan digunakan pada penelitian kali ini, yaitu Arduino Uno seperti pada Gambar 2.4. Arduino Uno adalah mikrokontroler berbasis ATmega 328P dengan *Clock Speed* 16Mhz dan *Flash Memory* 32 KB. Dapat berjalan pada catu daya 7 – 12 V. memiliki 14 pin digital *input/output* pada pin 0-13. 6 pin PWM pada pin 3, 5, 6, 9, 10, dan 11. 6 pin analog *input* pada pin A0 – A5, sambungan USB, sambungan catu daya tambahan dan tombol pengatur ulang.



Gambar 2.4 Arduino Uno



Gambar 2.5 GPRS Shield SIM 900

2.4 Modul SIM 900

Modul Sim 900 atau biasanya disebut dengan modul GPRS seperti pada Gambar 2.5 adalah suatu perangkat dari SIMCOM yang digunakan untuk berkomunikasi menggunakan jaringan GSM telepon. Perangkat ini mengizinkan pengguna untuk melakukan SMS, MMS, GPRS, dan audio menggunakan UART dengan mengirimkan perintah atau dalam hal ini *AT Command*. GPRS *Shield* SIM 900 ini memiliki 12 pin GPIO, 2 PWM, dan ADC. Dengan Fitur Quad – Band 850/900/1800/1900 MHz yaitu bisa digunakan pada jaringan GSM diberbagai Negara. Embedded TCP/UDP stack yang dapat upload data ke web server. Dan bekerja pada temperatur $-40^{\circ}\text{C} - 85^{\circ}\text{C}$ [7].

2.5 Pompa Air

Pompa air adalah suatu peralatan mekanik yang digerakkan oleh tenaga mesin yang digunakan untuk memindahkan cairan (*fluida*) dari suatu tempat ke tempat lain, dimana cairan tersebut hanya mengalir apabila terdapat perbedaan tekanan. Pompa juga dapat diartikan sebagai alat untuk memindahkan energi dari pemutar atau penggerak ke cairan ke bejana yang bertekanan yang lebih tinggi. Selain dapat memindahkan cairan pompa juga berfungsi untuk meningkatkan kecepatan, tekanan dan ketinggian cairan [8]. Tampilan pompa air pada Gambar 2.6

2.6 LED Growlight

LED (*Light emitting diode*) seperti pada Gambar 2.7 adalah suatu semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan. LED *Growlight* disini merupakan alat penerangan untuk pertumbuhan tanaman dengan menggunakan jenis LED berwarna merah dan biru yang dirangkai bersama.



Gambar 2.6 Pompa Air



Gambar 2.7 LED Growlight

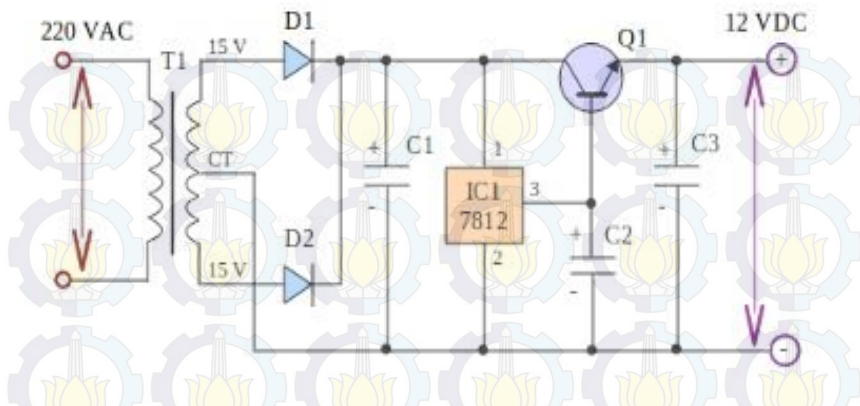
Karena LED berwarna merah memiliki panjang frekuensi ± 600 nm dan LED berwarna biru memiliki panjang frekuensi ± 400 nm sehingga dengan adanya LED *growlight* ini diharapkan waktu fotosintesis yang dilakukan tanaman hidroponik akan lebih lama karena adanya sumber cahaya dari LED dan tanaman akan lebih produktif karena akan mempercepat masa panen [8].

2.7 Relay

Rangkaian Relay adalah rangkaian elektronika yang digunakan untuk mengendalikan sesuatu dari jarak jauh. Relay sendiri merupakan saklar magnetis paling sering digunakan pada setiap rangkaian elektronika. Dalam dunia elektronika, relay sangat berperan penting dalam suatu rangkaian beban arus tinggi dengan arus rendah.

2.8 Power Supply

Power Supply atau pencatu daya merupakan rangkaian elektronika yang dapat menghasilkan energi listrik atau sebagai sumber energi untuk rangkaian elektronika lainnya. Sumber arus dari *power supply* adalah arus bolak – balik (AC) dari pembangkit listrik yang kemudian diubah menjadi arus searah (DC). Untuk dapat melakukan hal tersebut *power supply* memerlukan perangkat yang bisa mengubah arus AC menjadi DC. Untuk memperoleh tegangan DC asli memerlukan beberapa rangkaian pendukung lainnya yaitu Transformator 5 A, Dioda *Full – Wave Rectifier (Diode Bridge)*, *Voltage Regulator* LM7805, LM7809, LM7812, dan kapasitor. Contoh rangkaianannya seperti pada Gambar 2.8



Gambar 2.8 Rangkaian *Power Supply*

2.9 Transformator 5A

Transformator atau trafo adalah alat yang dapat berfungsi memindahkan tenaga listrik antar dua rangkaian listrik atau lebih melalui induksi elektromagnetik. Transformator bekerja berdasarkan prinsip induksi elektromagnetik. Tegangan masukan bolak – balik yang membentangi primer menimbulkan fluks magnet yang idealnya semua bersambung dengan lilitan sekunder. Fluks bolak – balik ini menginduksikan gaya gerak listrik (ggl) dalam lilitan sekunder. Jika efisiensi sempurna, semua daya pada lilitan primer akan dilimpahkan ke lilitan sekunder. Dalam rancangan sistem ini digunakan transformator 5 Ampere untuk menyuplai rangkaian pada sistem.

2.10 Voltage Regulator LM7805, LM7809, dan LM7812

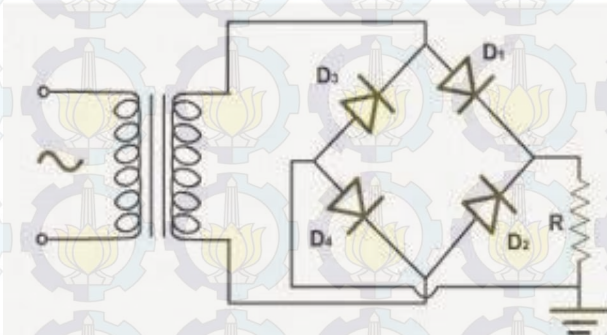
Voltage Regulator atau pengatur tegangan adalah salah satu rangkaian yang sering dipakai dalam peralatan elektronika. Fungsi *Voltage Regulator* adalah untuk mempertahankan atau memastikan tegangan pada level tertentu secara otomatis. Artinya, tegangan *output* (keluaran) DC pada *Voltage Regulator* tidak dipengaruhi oleh perubahan tegangan *input* (masukan), bebas pada keluaran dan juga suhu. Tegangan stabil yang bebas dari segala gangguan seperti *noise* ataupun fluktuasi (naik turun) sangat dibutuhkan untuk mengoperasikan peralatan elektronika. Pada sistem ini digunakan IC LM 7805, 7809, dan 7812 untuk menjadikan tegangan keluaran +5 volt, +9volt, dan +12 volt.

2.11 Kapasitor

Kapasitor adalah suatu komponen elektronika yang berfungsi untuk menyimpan arus listrik dalam bentuk muatan, selain itu kapasitor juga dapat digunakan sebagai penyaring frekuensi. Kapasitas untuk menyimpan kemampuan kapasitor dalam muatan listrik disebut Farad (F) sedangkan simbol dari kapasitor adalah C (Kapasitor). Sebuah kapasitor pada dasarnya terbuat dari dua buah lempengan logam yang saling sejajar satu sama lain dan diantara kedua logam tersebut terdapat bahan isolasi yang sering disebut dielektrik. Kapasitor berfungsi sebagai filter pada sebuah rangkaian *power supply*, maksudnya kapasitor sebagai *ripple* filter, berdasarkan sifat dasar kapasitor yaitu dapat menyimpan muatan listrik yang berfungsi untuk memotong tegangan *ripple*.

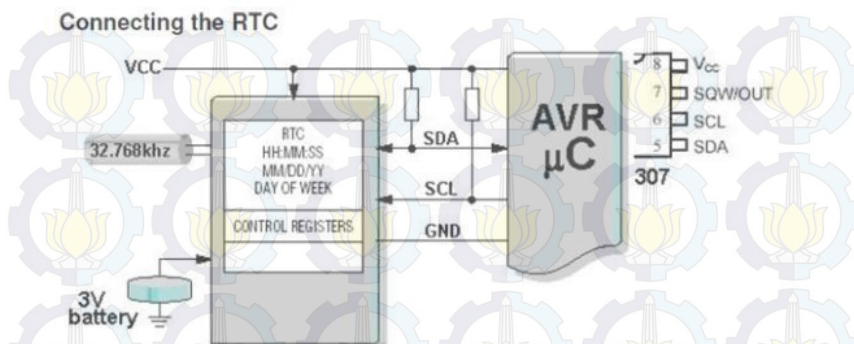
2.12 Dioda Full – Wave Rectifier (Diode Bridge)

Untuk penyearah gelombang penuh dengan 4 dioda menggunakan transformator non – CT seperti terlihat pada Gambar 2.9 Rangkaian *Diode Bridge*. Prinsip kerja penyearah gelombang dengan 4 dioda yaitu dimulai pada saat *output* transformator memberikan level tegangan sisi positif, maka D1, D4 pada posisi *forward* bias dan D2, D3 pada posisi *reverse* bias sehingga level tegangan sisi puncak positif tersebut akan dilewatkan melalui D1 ke D4. Kemudian pada saat output transformator memberikan level tegangan sisi negatif tersebut dialirkan melalui D2, D4.



Gambar 2.9 Rangkaian *Diode Bridge*

2.13 Real Time Clock (RTC)



Gambar 2.10 Rangkaian *Real Time Clock*.

RTC adalah sebuah rangkaian elektronika yang berfungsi sebagai acuan waktu[8]. Pada umumnya RTC digunakan pada alat elektronika yang membutuhkan akurasi waktu yang sesuai dengan waktu dunia. RTC berbeda dengan jam biasa karena RTC umumnya hanya dalam bentuk IC. Dalam penggunaannya, dengan adanya RTC sebuah sistem dapat fokus dengan tugas utamanya. Selain itu, RTC mempunyai sumber daya yang berbeda dari sistem. Sehingga ketika sistem dimatikan RTC masih berfungsi dan waktunya tidak akan berhenti atau *ter – reset* saat *restart*. Data – data yang tersimpan pada IC DS3232 disimpan pada register 00H untuk detik, 01H untuk menit, 02H untuk jam, 03H untuk hari, 04H untuk tanggal, 05H untuk bulan 06H untuk tahun, 07H untuk kontrol dan RAM 56x8 pada register 08H – 3FH.

Register tersebut bisa diakses oleh mikrokontroler melalui bus I2C. Gambar 2.10 merupakan gambar rangkaian dari RTC.

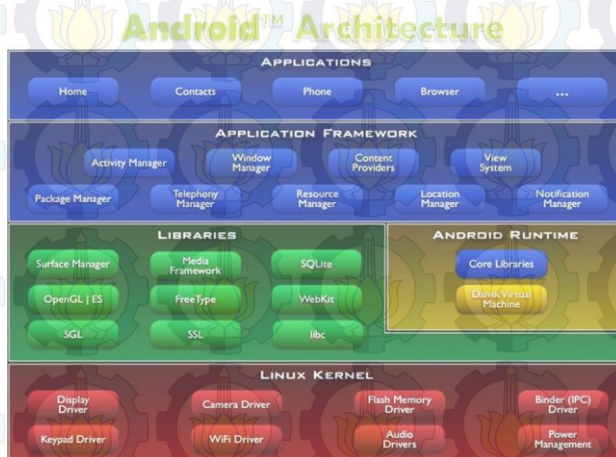
2.14 Sensor LDR (*Light Dependent Resistors*)

Sensor LDR berupa resistor yang dapat mengalami perubahan resistansi apabila mengalami perubahan penerimaan cahaya. Sensor ini terbuat dari *cadmium sulfide* yaitu merupakan bahan semikonduktor yang resistansinya berubah – ubah menurut banyaknya cahaya sinar yang mengenainya[10].

2.15 Sistem Operasi Android

Sistem operasi android tersusun atas elemen – elemen seperti *application layer*, *application framework*, *libraries*, dan *Linux kernel*. Kernel linux adalah layanan inti (termasuk didalamnya *driver hardware*, manajemen proses dan memori, keamanan, *network*, dan manajemen daya) yang dikerjakan oleh kernel Linux 2.6. kernel juga menyediakan lapisan abstraksi (*abstraction layer*) antara *hardware* dan elemen lainnya. Pustaka (*library*) berjalan pada bagian atas dari kernel, termasuk didalamnya berbagai macam pustaka inti seperti pustaka (*library*) media yang digunakan untuk memainkan audio dan video, *surface manager* untuk menyediakan manajemen display, pustaka grafik (*Graphics libraries*) yang didalamnya SGL dan OpenGL untuk grafis 2D dan 3D, SQLite untuk layanan basis data, SSL dan Webkit untuk web browser dan keamanan internet (*Internet Security*) terintegrasi[11].

Gambar 2.11 merupakan Arsitektur dari sistem operasi android. Android *Runtime* merupakan mesin yang akan menjalankan tiap – tiap aplikasi yang ada. Android *runtime* tersusun atas 2 elemen yaitu Pustaka inti (*Core Library*) yang merupakan Pengembangan android dengan menggunakan java dan Dalvik *Virtual Machine* yang telah dioptimalisasi agar perangkat dapat menjalankan berbagai tugas dengan efisien.



Gambar 2.11 Arsitektur Sistem Operasi Android

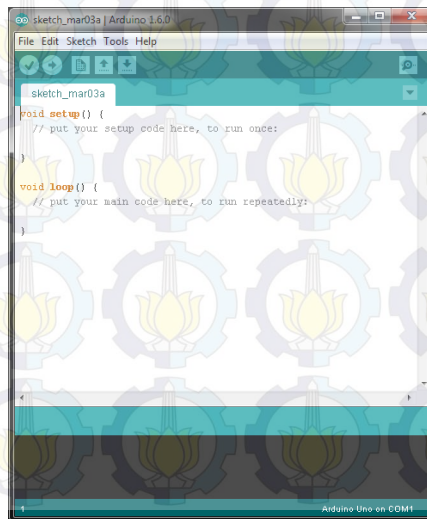
Application framework menyediakan *class – class* yang dapat digunakan ketika membuat aplikasi android. *Application framework* juga menyediakan abstraksi generik dari perangkat keras dan mengelola antarmuka serta *resource* dari aplikasi. *Application layer* berisi tentang aplikasi pihak ketiga dan menggunakan pustaka (*libraries*) API yang sama.

2.16 LCD QC1602A

QC1602A adalah modul tampilan LCD (*Liquid Crystal Display*) dengan spesifikasi yaitu layar 2.6” *positive transfective* dengan 16 karakter 2 baris (putih), LED *backlight* (biru), 4/8 bit *parallel interface*, tegangan kerja 5 volt dan 16 pin[12]. Gambar 2.12 merupakan tampilan dari modul LCD QC1602A



Gambar 2.12 Modul LCD QC1602A



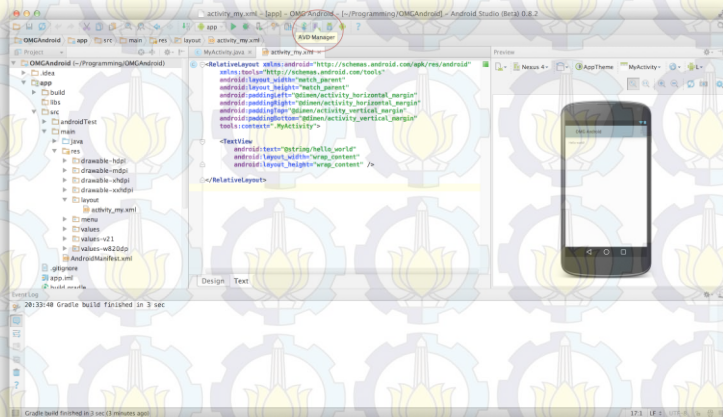
Gambar 2.13 Tampilan Arduino IDE

2.17 Arduino IDE

Software arduino atau biasa disebut dengan arduino IDE (*Integrated Development Environment*) digunakan untuk memprogram arduino. *Software* arduino ini dapat berjalan di semua *operating system* seperti Windows, Mac, dan Linux. Berdasarkan pengolahan, avr – gcc dan perangkat lunak sumber terbuka lainnya. Perangkat lunak Arduino IDE (*Integrated Development Environment*) dapat diunduh dengan mudah di *website* resmi Arduino. Sifat dari Arduino IDE yang *open – source* mengakibatkan pemakaian dari perangkat lunak ini dapat digunakan secara bebas dan selalu mendapatkan pembaruan dari Komunitas sehingga penggunaan perangkat lunak ini semakin lama semakin baik. Tampilan dari Arduino IDE seperti pada Gambar 2.13

2.18 Android Studio

Dalam memprogram aplikasi android salah satu *software* yang digunakan adalah android studio. Untuk membuat aplikasi android dengan android studio menggunakan bahasa pemrograman Java dan untuk membuat tampilannya dengan XML (*Extensible Markup Language*). Untuk bisa menggunakan *software* ini disarankan RAM dari komputer yang digunakan adalah 4 GB. Android studio dapat digunakan di sistem operasi linux, windows, dan mac [13]. Gambar 2.14 merupakan tampilan dari perangkat lunak android studio



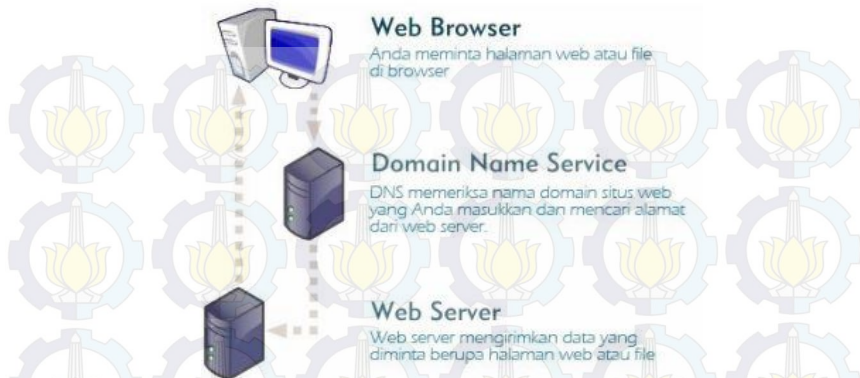
Gambar 2.14 Tampilan *Software* Android Studio

2.19 Sensor YL – 69

Sensor YL – 69 merupakan sensor air sederhana yang dapat digunakan untuk mendeteksi kelembaban tanah dan tingkat kejernihan air [14]. Nilai yang dibaca oleh sensor kelembaban YL – 69 menghasilkan nilai yang besar pada tanah dengan kandungan air yang lebih banyak. Sensor YL – 69 merupakan sensor yang membaca nilai kelembaban berdasarkan konstanta dielektrik yang diukur dengan *transmission – line technique* saat dialiri listrik oleh lengan sensor [15]. Sensor ini dapat mendeteksi kadar air dalam yang kemudian bisa menjadi dalam sistem pengairan. Untuk pendeteksian secara presisi menggunakan arduino, dapat menggunakan pin keluaran analog (sambungkan dengan pin ADC atau *Analog – input converter*). Modul pemroses dapat menggunakan catu daya antara 3,3 volt hingga 5 volt sehingga fleksibel untuk digunakan pada berbagai macam mikrokontroler.

2.20 Web Hosting

Web Hosting atau bisa disebut dengan *Virtual Hosting* merupakan kumpulan dari beberapa *server* yang menyediakan *service – service* guna memberikan kemudahan bagi pengguna untuk membuat *website* [16]. Dengan kata lain *Virtual Hosting (Web Hosting)* adalah penyedia layanan untuk menampung data – data yang diperlukan oleh sebuah *website* dan sehingga dapat diakses lewat internet. Data disini dapat berupa file, gambar, email, aplikasi/program/script dan *database*. Terdapat 2 jenis *hosting* pada sistem operasi, yaitu *hosting* pada sistem operasi *Windows* dan *Unix*. Pada sistem *Unix*, sistem operasinya dianggap lebih serta cepat karena mendukung *open-source* program *freeware* dan *shareware*. *Web hosting* berbasis *Unix* sangat baik bagi pengguna yang menginginkan tingkat keamanan yang tinggi [17]. *Virtual hosting* menyediakan layanan tertentu, tentunya yang mendukung keperluan membangun sebuah *website*. Layanan – layanan yang disediakan oleh *Virtual Hosting* antara lain adalah *HTTP server* atau disebut juga dengan *Webserver*, *database server*, *FTP server*, menyimpan semua *inbox* mereka dalam *mail server* tanpa takut *mailbox* penuh atau membuat *website* pribadi sesuai keinginan melalui layanan *webserver* [16]. Saat mengakses sebuah halaman web pada dasarnya kita sedang download dari *web server* ke komputer lokal.



Gambar 2.15 Gambaran Umum *Web Hosting*

Komponen utama dalam *website* agar bisa diakses dengan mengetikkan alamat *website* (*URL*) adalah domain dan hosting. Domain adalah nama *URL* tersebut. Sedangkan hosting adalah penyedia layanan untuk menampung data – data yang diperlukan oleh sebuah *website* sehingga dapat diakses melalui internet. Gambaran umum dari *web hosting* seperti pada Gambar 2.15

2.21 Database Server

Database *Server* adalah sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan model *client – server* [16]. Istilah ini juga merujuk kepada sebuah komputer (umumnya merupakan *server*) yang diperuntukan untuk menjalankan program basis data. Sistem manajemen basis data pada umumnya menyediakan fungsi – fungsi *server* basis data. Beberapa manfaat yang diberikan oleh database *server* antara lain adalah meningkatkan pencarian dan pengambilan data, menambah tingkat keamanan data, dan dioptimasi untuk melayani permintaan dari banyak *user* dan tidak terpengaruhi dengan besarnya data yang telah ada.

2.22 JSON

Javascript Object Notation (*JSON*) adalah format pertukaran data yang ringan, *text-base*, format yang mudah dibaca untuk merepresentasikan struktur data dan objek [18]. *JSON* merupakan

format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari keluarga – C, termasuk C, C++, C#, Java, Javascript, Perl, Python, dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data yang ideal. Untuk membuat dan mengolah JSON dengan bahasa pemrograman web, yaitu PHP tidak perlu lagi menambahkan modul atau *plugin* karena PHP sudah mendukung format JSON seperti Gambar 2.16 yaitu menggunakan fungsi `json_decode()` atau `json_encode()`.

Penggunaan JSON *parsing* dalam HTTP *Connection* ini bertujuan untuk memberi kemudahan bagi *user* ketika menggunakan aplikasi lain. Dengan adanya JSON *parsing* HTTP *Connection*, informasi yang ada didalam *website* dapat ditampilkan dalam sebuah aplikasi *mobile* [19]. Hal ini menyebabkan aplikasi *mobile* tidak perlu menampilkan seluruh *content* yang ada didalam *website* seperti halnya pada *mobile web browser*. Aplikasi android akan memilih jenis konten yang ingin ditampilkan, seperti halnya gambar, *item description*, dan lain – lain. Dalam JSON *parsing*, kita membutuhkan API (*Application Program Interface*) yang berfungsi untuk menghubungkan antara aplikasi *mobile* dan aplikasi *website* [19].

JSON dengan 2 element	JSON dengan 2 kedalaman
<pre>{ "data": { "element": [{ "item1": "aaaaaaaaa" "item2": "bbbbbbbt" }] } }</pre>	<pre>{ "data": { "element1": { "element": { "item1": ["aaaaaaaaa" "bbbbbbbt"] } } } }</pre>

Gambar 2.16 Contoh Data JSON

2.23 API (*Application Program Interface*)

Pada penelitian ini API ialah kumpulan *source code* berbasis PHP, yang isinya adalah Query untuk mengambil data dari *website*, yang hasilnya di – *encode* ke dalam bentuk JSON. Pada saat aplikasi *mobile* mengirimkan *request* untuk menampilkan data (*SELECT * FROM*) dari *website*, API akan meneruskannya sesuai dengan Query yang tepat untuk menampilkan data (*SELECT * FROM*). Setelah data diambil, kemudian di – *encode* ke dalam bentuk JSON, dan diteruskan ke dalam aplikasi *mobile*. Di dalam aplikasi *mobile*, JSON tersebut di – *parsing* ke dalam bentuk *list*. Gambar 2.17 menunjukkan arsitektur JSON *parsing*.



Gambar 2.17 Arsitektur JSON *Parsing*



BAB III

PERANCANGAN SISTEM KONTROL

Pada bab ini akan dibahas mengenai perancangan alat yang meliputi perancangan perangkat keras (*hardware*) dan perangkat lunak (*software*). Hal ini berguna mewujudkan Tugas Akhir yang berjudul “Rancang Bangun Sistem Otomasi Pemberian Nutrisi dan Pencahayaayaan untuk Tahap Penyemaian Benih Selada pada Perkebunan Surabaya Hidroponik”. Perancangan alat akan dibahas per – bagian disertai dengan gambar skematik. Sedangkan penjelasan *software* akan dijelaskan mengenai pembuatan program sistem kontrol menggunakan Arduino dan program *interfacing* menggunakan Sistem Operasi Android agar nantinya sistem hidroponik dapat dimonitoring dengan baik oleh *user*.

Untuk memudahkan dalam pembahasan bab ini akan dibagi menjadi dua yaitu:

1. Perancangan *hardware* (perangkat keras) yang terdiri dari perancangan rangkaian kontroler Arduino Mega 2560, perancangan rangkaian *power supply*, perancangan rangkaian pengondisi sinyal sensor EC, perancangan rangkaian pengondisi sinyal untuk LDR, pembuatan rangkaian RTC (*Real Time Clock*), perancangan rangkaian *driver relay*, perancangan LED *Grow light*, perancangan rangkaian pengatur intensitas *grow light*, perancangan *hardware* pompa nutrisi dan tangki nutrisi, perancangan *hardware* untuk sistem pengarian dengan pompa, perancangan indikator. Serta perancangan *hardware plant* yang terdiri dari *prototype* rak tanaman hidroponik dan kerangka bak reservoir.
2. Perancangan *software* (perangkat lunak) meliputi perancangan *Virtual Hosting* , pembuatan database *server phpmyadmin* , pembuatan program modul GPRS SIM 900 dan kontrol dengan Arduino, dan perancangan pembuatan program *client* pada sistem operasi android dengan menggunakan *software* Android Studio.

3.1 Perancangan Hardware (Perangkat Keras)

Pada sistem perancangan perangkat keras yang akan dibuat, RTC (*Real Time Clock*) memiliki peran yang sangat penting dimana

komponen ini sebagai acuan untuk mengatur penjadwalan perawatan tanaman. Ketika umur tanaman dibawah 7 hari maka mikrokontroler akan mengaktifkan pompa bak reservoir setiap hari pada pukul 07.00 – 17.00 WIB selama 5 menit. Ketika umur tanaman lebih dari 7 hari dan kurang dari 15 hari maka sensor EC akan mendeteksi kadar nutrisi pada bak reservoir kemudian hasil dari pembacaan sensor ini diterima oleh mikrokontroler. Kemudian mikrokontroler akan melakukan penanganan terhadap kondisi nutrisi yang ada pada bak reservoir. Apabila pada kadar nutrisi pada bak reservoir kurang dari 400 ppm maka pompa nutrisi akan aktif. Jika sensor EC mendeteksi kepekatan nutrisi diantara 400 – 900 ppm maka pompa nutrisi akan mati. Dan jika lebih dari 900 ppm maka alarm akan aktif. Skema prinsip kerja sistem yang akan di buat dapat dilihat pada Gambar 3.1.

Berdasarkan Gambar 3.1 untuk mengatur intensitas pencahayaan, sistem ini menggunakan LDR (*Light Dependent Resistor*) sebagai sensornya. Ketika tanaman kekurangan cahaya, LDR akan mengeluarkan tegangan tertentu yang diterima mikrokontroler kemudian akan mengaktifkan *growlight* dengan intensitas tertentu. Dengan sistem seperti ini diharapkan dapat menghasilkan bibit tanaman hidroponik yang baik dan siap dibudidayakan.

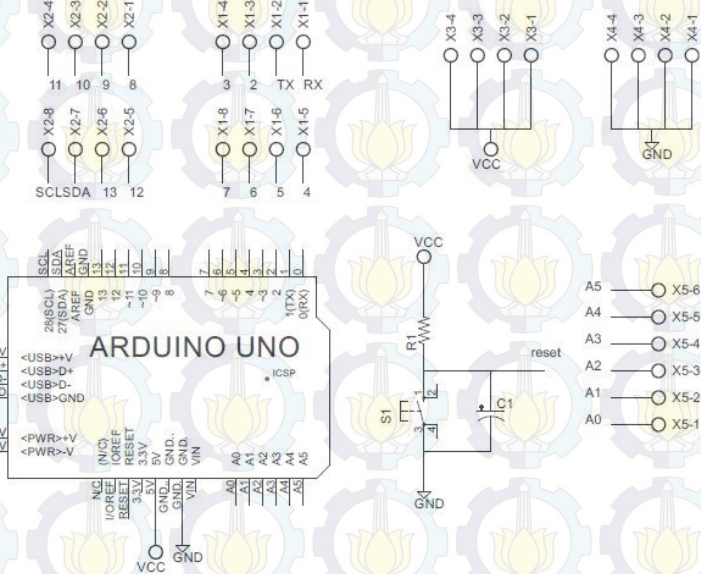


Gambar 3.1 Skema Prinsip Kerja Sistem

Sedangkan untuk memonitoring kadar nutrisi oleh *user* sistem ini menggunakan SIM 900 sebagai media pengirim data ke database *server* kemudian data tersebut ditampilkan pada sebuah aplikasi di *smartphone* dengan sistem operasi android.

3.2 Perancangan Kontroler Arduino Uno

Dalam perancangan rangkaian kontroler ini digunakan Arduino UNO sebagai kontroler untuk menerima keadaan sensor EC pada bak reservoir, menerima keadaan sensor LDR, menerima data waktu dari RTC, dan menjalankan program utama dan mengirim data ke *server* melalui modul SIM 900. *Software* yang digunakan dalam pembuatan program untuk Arduino UNO adalah Arduino IDE. Pada kontroler Arduino UNO ini di tambahkan *shield* tambahan. Pada shield ini terdapat terminal. Terminal-terminal ini terhubung dengan pin-pin Arduino UNO yang di gunakan untuk mendukung berjalannya sistem. Tujuan dibuat terminal ini adalah untuk memudahkan wiring dan agar terlihat rapi.



Gambar 3.2 Skematik Shield Tambahan

Terminal-terminal ini terbagi menjadi 4 bagian yaitu terminal power, terminal input analog, terminal output digital (PWM) dan terminal komunikasi. Dalam shield ini terdapat 4 terminal 5V dan 4 terminal GND. Sedangkan untuk terminal input analog terdapat 8 terminal yaitu A0, A1, A2, A3, A4 dan A5. Untuk output digital terdapat 12 terminal. Dan untuk komunikasi terbagi menjadi 2. Yang pertama yaitu komunikasi serial pada terminal TX dan RX. Dan yang kedua terdapat komunikasi I2C (*Inter Integrated Circuit*) pada terminal SDA dan SCL. Arduino ini deprogram agar dapat mengontrol kadar EC pada bak reservoir yang mengalir ke *plant*. Untuk menjaga level dari nutrisi pada tempat penampung nutrisi. Menyalakan dan mematikan LED *growlight*, mengatur PWM yang masuk ke rangkaian LED *grow light* dan mengatur pengiriman data menggunakan modul SIM 900. Skematik dari *shield* tambahan dapat dilihat pada Gambar 3.2.

Untuk lebih jelasnya bisa lihat di Tabel 3.1 untuk konfigurasi pada *shield* tambahan.

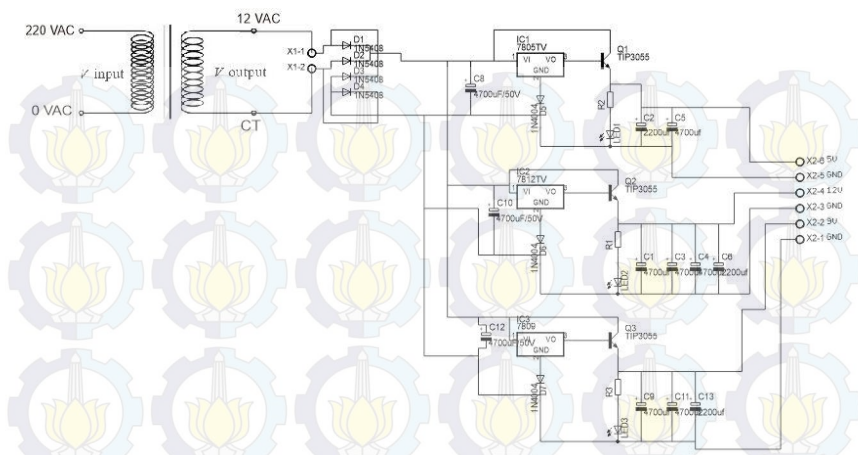
Tabel 3.1 Konfigurasi Port *Shield* Tambahan

No.	Pin Arduino	Keterangan
1.	Pin A0	Analog input sensor EC
2.	Pin A1	Analog input LDR
3.	Pin 2	Output digital relay pompa nutrisi B
4.	Pin 3	Output PWM LED <i>Growlight</i>
5.	Pin 4	Output digital relay LED <i>Growlight</i>
6.	Pin 5	Output digital relay pompa nutrisi A
7.	Pin 6	Indikator EC kurang dari normal
8.	Pin 7	Indikator EC dalam kondisi normal
9.	Pin 8	Indikator EC lebih dari kondisi normal
12.	Pin 12	Pengaduk pada bak reservoir
13.	Pin 13	Pompa pada bak reservoir
12.	Pin SDA	Input RTC dan LCD SDA
13.	Pin SCL	Input RTC dan LCD SCL
14.	Pin 5 V	Output 5 V
15.	Pin GND	Output 0 V

Pin-pin yang di gunakan untuk mendukung berjalannya sistem adalah pin A1 digunakan untuk analog input LDR (*Light Dependent Resistor*), pin A0 digunakan untuk Analog input sensor EC (*Electrical Conductivity*), Pin 2 digunakan untuk digital input relay pompa nutrisi B, pin 3 digunakan untuk output PWM LED *Growlight*, pin 4 digunakan untuk digital input relay LED *Growlight*, pin 5 digunakan untuk digital input relay pompa nutrisi A, pin 6 digunakan untuk indikator ketika EC kurang dari kondisi normal, pin 7 digunakan untuk indikator ketika EC berada pada kondisi normal , pin 8 digunakan sebagai indikator ketika EC lebih dari kondisi normal, pin 12 digunakan sebagai digital input relay motor pengaduk, pin 13 di gunakan sebagai digital input relay pompa bak reservoir, pin SDA digunakan untuk input RTC dan LCD SDA, dan pin SCL digunakan untuk input RTC dan LCD SCL.

3.3 Perancangan Power Supply

Pada perancangan power supply ini akan direalisasikan output tegangan sebesar 5V, 9V dan 12V. Tegangan 5V digunakan untuk mencatu pompa nutrisi dan. Sedangkan tegangan 9V di gunakan untuk mencatu rangkaian pengondisi sinyal sensor EC dan LED *grow light*. Sedangkan untuk tegangan 12V dignakan untuk mencatu rangkaian driver relay. Pada output tegangan 5V cukup menggunakan 2 buah kapasitor dengan kapasitansi 2200uF dan 4700uF karena tegangan 5V tidak banyak di gunakan untuk mencatu rangkaian sehingga kemungkinan untuk drop tegangan kecil. Sedangkan pada output 9V digunakan 3 buah kapasitor aitu 2 buah kapitor sebesar 4700uF dan 1 kapasitor sebesar 2200 uF yang berfungsi untuk meng – *cover* tegangan agar tidak terjadi drop tegangan ketika digunakan untuk mencatu rangkaian pengondisi sinyal sensor Ecdan LED *grow light*. Pada jalur tegangan +12 volt digunakan 4 buah kapasitor dengan 3 buah kapasitor dengan kapasitansi sebesar 4700 uF dan 1 buah kapasitor dengan kapasitansi 2200uF yang berguna untuk meng – *cover* tegangan pada rangkaian tersebut jika terjadi drop tegangan yang dikarenakan beban (*solenoid push – pull*) yang mengalami *overshoot* pada saat pertama kali dijalankan karena tegangan 12 volt ini digunakan untuk rangkaian driver relay. Pada rangkaian ini juga dilengkapi dengan komponen *power transistor* (TIP 3055) yang berfungsi sebagai penguatan arus ketika mendapat beban yang besar. Untuk menentukan berapa besar arus yang dikeluarkan oleh sistem, maka digunakan hokum *Kirchoff* arus atau KCL (*Kirchoff Current Law*) yang berbunyi :



Gambar 3.3 Skematik Rangkaian *Power Supply*

“Jumlah arus yang mengalir masuk ke sebuah *node* (titik percabangan) akan sama dengan jumlah arus yang keluar dari *node* tersebut”, atau dengan kata lain jumlah arus disetiap percabangan rangkaian sama dengan jumlah arus yang dikeluarkan pada jalur awal *power supply*, dengan rumus:

$$I_s = \sum I_n \dots \dots \dots (3.1)$$

Keterangan:

n = jumlah percabangan untuk arus yang masuk

I_s = besar arus yang dikeluarkan oleh sumber

I_n = jumlah total arus pada setiap percabangan

Sedangkan untuk menjalankan fungsi dari prototipe hidroponik ini digunakan tegangan AC sebesar 220 V yang mengambil tegangan langsung dari sumber AC PLN. Gambar skematik rangkaian *power supply* dapat dilihat pada Gambar 3.3.

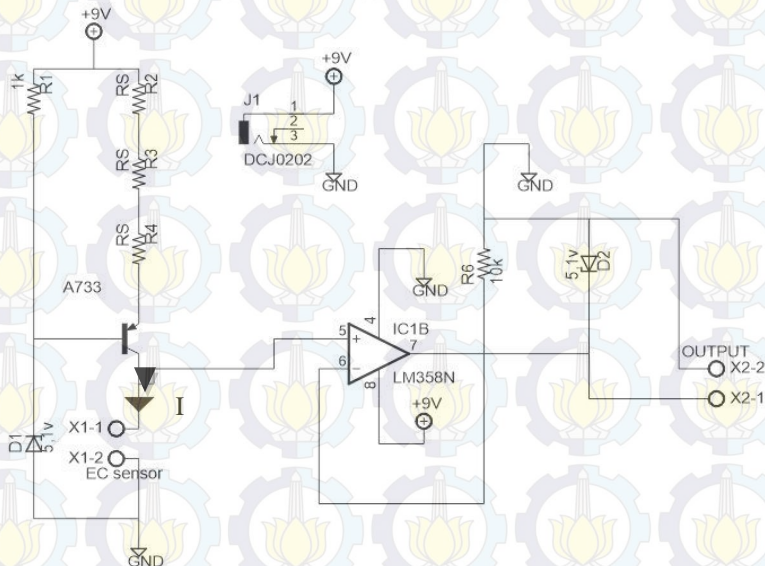
3.4 Perancangan Rangkaian Pengondisi Sinyal Sensor EC

Rangkaian pengondisi sinyal sensor EC ini terbagi menjadi 2 bagian. Pertama adalah rangkaian sumber arus dan yang kedua adalah rangkaian buffer. Gambar skematik rangkaian pengondisi sinyal sensor EC dapat dilihat pada Gambar 3.4. Fungsi dari rangkaian sumber arus

digunakan untuk mengalirkan arus yang stabil pada sensor EC. Output sensor EC berupa resistansi. Jika arus yang mengalir melalui sensor EC tidak stabil maka akan menghasilkan tegangan yang berubah-ubah. Hal ini mengakibatkan sensor menjadi tidak linier. Komponen yang diperlukan untuk rangkaian sumber arus adalah resistor 1K, dioda zenner 5,1V, transistor NPN A733. Nilai RS ($R_2 + R_3 + R_4$) dapat di cari dengan menggunakan persamaan :

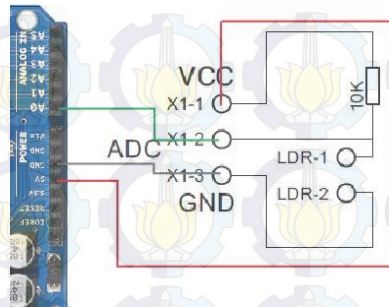
$$I = \frac{V_{cc} - V_{ce}}{R_2 + R_3 + R_4} \dots\dots\dots (3.2)$$

Sedangkan rangkaian buffer digunakan untuk menguatkan arus. Karena jika mengambil secara langsung tegangan pada sensor EC maka arus akan terbagi menjadi dua yaitu arus yang mengalir ke sensor dan ke mikokontroler. Untuk mencegah hal ini maka diperlukan rangkaian buffer. Untuk membuat rangkaian buffer komponen yang di gunakan adalah op-Amp LM358, resistor 10K ohm dan dioda zenner 5,1V.

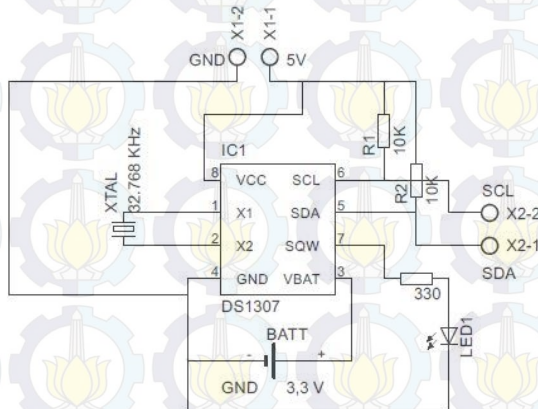


Gambar 3.4 Skematik Rangkaian Pengondisi Sinyal Sensor EC

3.5 Perancangan Rangkaian Pengondisi Sinyal LDR



Gambar 3.5 Skematik Rangkaian Pengondisi Sinyal LDR



Gambar 3.6 Skematik Rangkaian RTC

LDR merupakan salah satu dari resistor variabel. LDR sangat peka terhadap rangsangan cahaya. Output dari LDR berupa resistansi. Semakin besar intensitas cahaya yang diterima LDR maka resistansinya semakin kecil. Sebaliknya jika intensitas cahaya yang diterima LDR kecil (gelap) maka resistansinya semakin besar. Rangkaian pengondisi sinyal LDR menggunakan rangkaian pembagi tegangan. Skematik rangkaian rangkaian pengondisi sinyal LDR dapat dilihat pada Gambar 3.5. Fungsi dari rangkaian pengondisi sinyal LDR ini adalah untuk mendeteksi intensitas cahaya yang diterima oleh tanaman. Jika tanaman

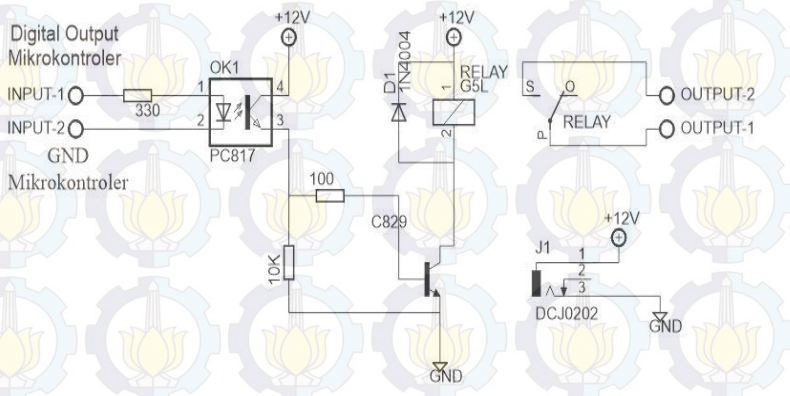
kurang menerima intensitas cahaya maka rangkaian ini akan mengirimkan tegangan terhadap mikrokontroler dan mikrokontroler akan melakukan penanganan berupa mengaktifkan LED *grow light*.

3.6 Perancangan RTC (*Real Time Clock*)

RTC adalah jam elektrik berupa chip yang dapat menghitung mulai detik sampai tahun. RTC merupakan rangkaian penting pada sistem ini karena digunakan untuk penjadwalan perawatan terhadap penyemaian bibit. Gambar 3.6 merupakan skematik rangkaian RTC. Rangkaian RTC ini menggunakan IC DS 1307 sebagai komponen utamanya.

3.7 Perancangan Driver Relay

Pada rangkaian driver relay (Gambar 3.7) ini digunakan relay 12 volt, IC *optocoupler* PC817, transistor C829, resistor 330 ohm, 100 ohm, dan 1K ohm. IC *optocoupler* (PC817) yang berfungsi sebagai proteksi arus, agar arus yang digunakan untuk men – *trigger* transistor C829 tidak dapat bercampur atau dengan kata lain merusak komponen Arduino Mega 2560. Sedangkan *transistor* C829 digunakan untuk proses *switching* relay 12V. Driver relay yang digunakan pada sistem ini sebanyak 5 buah dimana masing – masing digunakan untuk saklar untuk menyalakan pompa nutrisi A dan B, sebagai saklar motor pengaduk, kemudian untuk saklar dari LED *grow light*, dan yang terakhir untuk saklar dari pompa bak reservoir.



Gambar 3.7 Rangkaian Driver Relay

Syarat transistor sebagai saklar adalah jika $I_c \leq \beta$ dan $\beta (hfe) \times I_b$. Besar I_c pada transistor C829 adalah 30 mA. Sedangkan $\beta (hfe)$ pada transistor C829 sebesar 70. I_b pada rangkaian di atas dapat dihitung dengan menggunakan rumus :

$$I_b = \left(\frac{V_{cc} - V_{be(c829)} - V_{ce(optocoupler)}}{R_b} \right)$$

$$I_b = \left(\frac{12V - 5V - 5V}{100} \right)$$

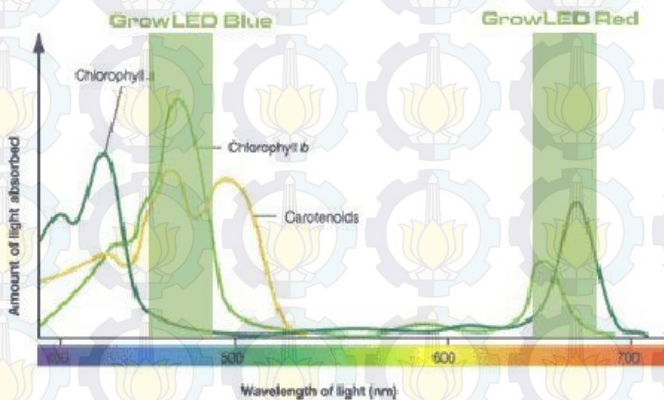
$$I_b = \frac{2}{100}$$

$$I_b = 0.02A$$

Maka :

$$\beta \times I_b = 70 \times 0.02A = 1.4A$$

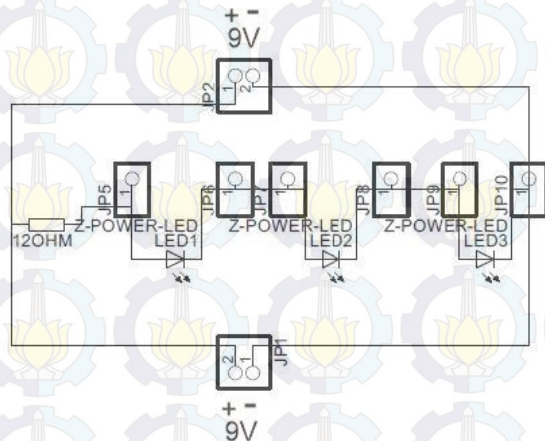
Sehingga syarat transistor C829 sebagai saklar terpenuhi.



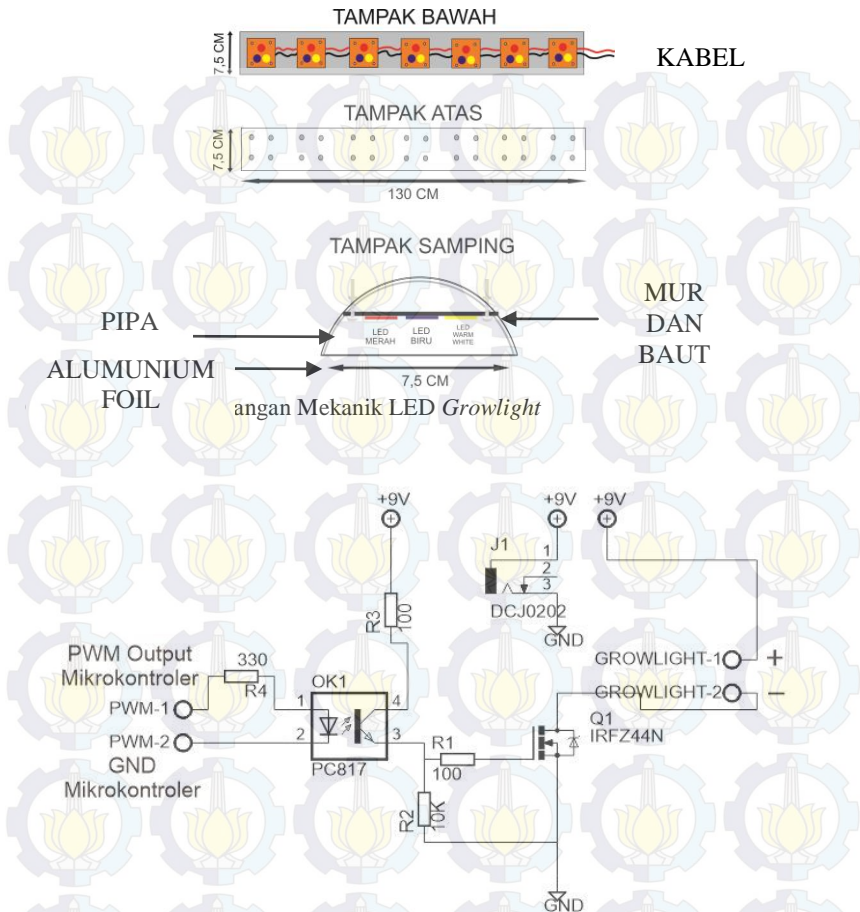
Gambar 3.8 Panjang Gelombang Spektrum Warna LED

3.8 Perancangan LED Growlight

Dalam perancangan LED grow light terbagi menjadi dua yaitu perancangan elektrik dan perancangan mekanik. Fungsi dari perancangan elektrik adalah untuk membuat sebuah rangkaian yang tidak banyak memakai tempat sehingga sesuai dengan *plant object* perkebunan hidroponik. Selain itu terdapat latar belakang bahwa tanaman berfotosintesis memanfaatkan cahaya dengan panjang gelombang 445nm (identik dengan spektrum gelombang cahaya warna merah) dan panjang gelombang 630nm - 660nm (identik dengan spektrum gelombang cahaya warna biru) seperti terlihat pada Gambar 3.8. Jadi untuk perancangan elektriknya LED *grow light* membutuhkan rangkaian yang terdiri dari 3 LED *super brihgt* 1 watt dengan warna merah, biru, *warm white* dan resistor 12 ohm $\frac{1}{2}$ watt. Fungsi dari perancangan mekanik adalah agar LED grow light terlihat kokoh dan mudah dalam instalasinya. Bahan yang di butuhkan dalam perancangan mekanik adalah pipa dengan diameter 7,5cm, alumunium *foil*, mur dan baut dengan panjang 2 cm masing-masing sebanyak 84 buah. LED grow light yang akan dibuat sebanyak 3 buah. Skematik rangkaian grow light dapat dilihat pada Gambar 3.9. Dan untuk perancangan mekaniknya dapat di lihat pada Gambar 3.10.



Gambar 3.9 Skematik Rangkaian Growlight



Gambar 3.11 Skematik Rangkaian Pengatur Intensitas *Growlight*

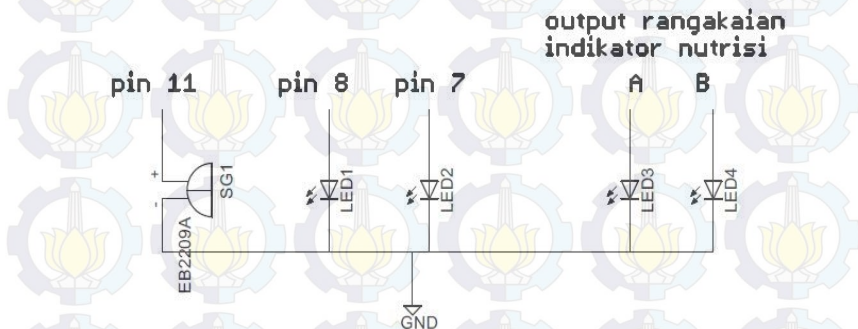
3.9 Perancangan Rangkaian Pengatur Intensitas *Growlight*

Rangkaian pengatur intensitas grow light berfungsi sebagai aktuatur dalam mengaktifkan *grow light* dengan intensitas tertentu. Perancangan ini bertujuan untuk mengatur intensitas cahaya yang diterima oleh tanaman jika tanaman kekurangan cahaya. Rangkaian ini menggunakan transistor IRF Z44N sebagai komponen utamanya. Transistor ini berfungsi sebagai *fast switching* yang *di-trigger*

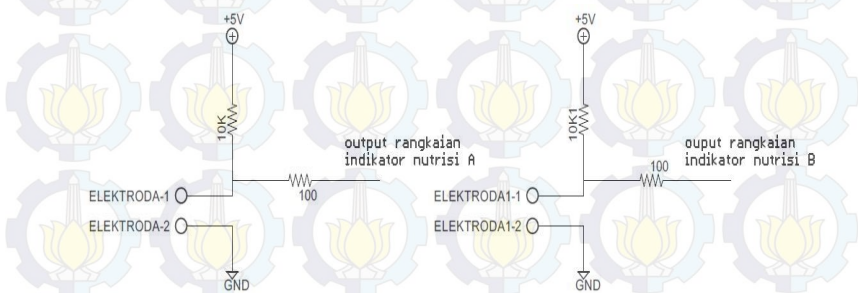
menggunakan PWM. Gambar skemaik dar rangkaian ini dapat di lihat pada Gambar 3.11.

3.10 Perancangan Rangkaian Indikator

Rangkaian indikator berfungsi sebagai mini interface dengan user berupa rangkaian LED dan *buzzer* yang mengindikasikan status sistem. Skematik rangkaian indikator daat dilihat pada Gambar 3.12. Buzzer akan aktif apabila kadar nutrisi (EC) melebihi batas normal. LED 1 digunakan untuk mengindikasikan kadar nutrisi (EC) dalam keadaan normal. LED 2 digunakan untuk mengindikasikan kadar nutrisi (EC) kurang dari batas normal. LED 3 dan LED 4 digunakan untuk mengindikasikan status nutrisi dalam tangki nutrisi masih ada atau sudah habis. Perbedaan dari LED 3 dan 4 adalah LED 3 untuk nutrisi A dan LED 4 untuk nutrisi B.

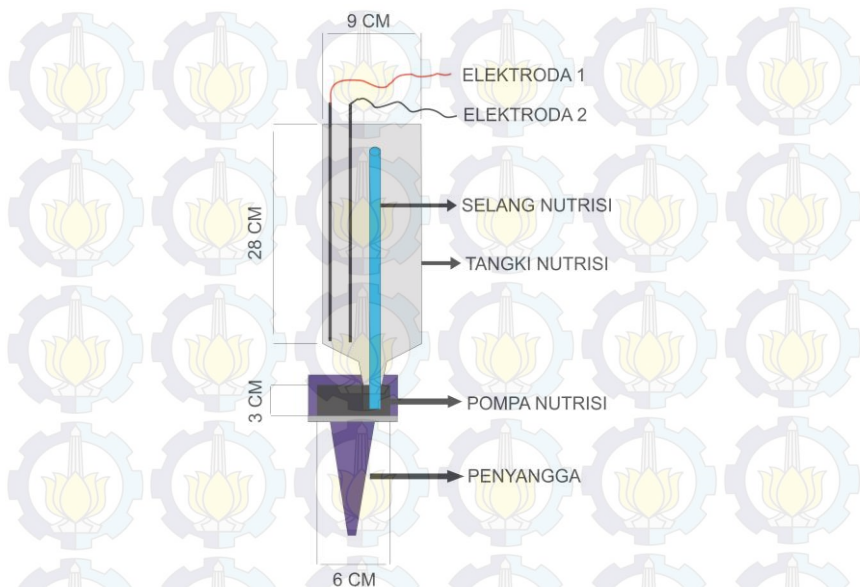


Gambar 3.12 Skematik Rangkaian Indikator



Gambar 3.13 Skematik Rangkaian Level Nutrisi

3.11 Perancangan Mekanik Pompa Nutrisi



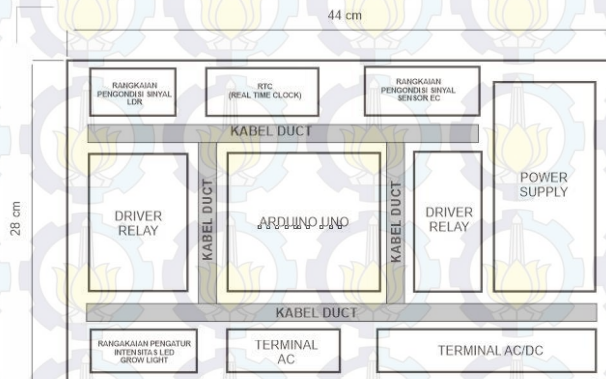
Gambar 3.14 Perancangan Mekanik Pompa Nutrisi

Pada perancangan kali ini terbagi menjadi dua yaitu perancangan elektrik da perancangan mekanik. Perancangan elektrik dibuat untuk dapat memonitoring status nutrisi dengan baik menggunakan sensor level nutrisi. Sedangkan perancangan mekanik di buat untuk menggabungkan antara pompa nutrisi dan tangki nutrisi. Skema rangkaian elektrik sensor level nutrisi dapat dilihat pada Gambar 3.13 . Sedangkan rancangan mekanik dari pompa nutrisis dan tangki nutrisi dapat di lihat pada Gambar 3.14

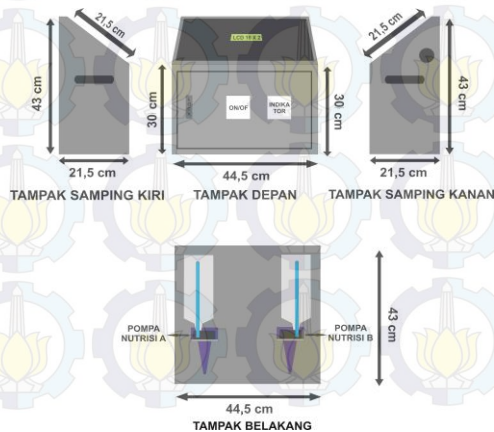
3.12 Perancangan *Hardware* Panel Kontrol

Perancangan *hardware* kontrol ini berfungsi sebagai tempat peletakan seluruh rangkaian yang disusun menjadi satu kesatuan agar terlihat rapi. Pada *hardware* kontrol ini terdiri dari *power supply*, *driver relay*, rangkaian pengondisi sinyal sensor EC, rangkaian pengondisi sinyal LDR, Arduino Mega 2560, terminal AC, terminal DC, dan rangkaian pengatur intensitas LED *growlight*. Seperti Gambar 3.15

desain dari perancangan *hardware* kontrol dalam panel. Adapun desain *hardware* kontrol yang terlihat dari bagian luar panel yaitu bagian depan panel, bagian belakang panel, samping kiri dan samping kanan dari panel. Gambar 3.16 merupakan desain *hardware* kontrol yang terlihat dari bagian depan, bagian belakang, samping kanan dan samping kiri. Pada panel tampak depan terlihat *lock* (pengunci), saklar *ON/OFF*, indikator dan LCD 16x2. Pada bagian belakang panel terdapat 2 pompa nutrisi dan penampung nutrisi.



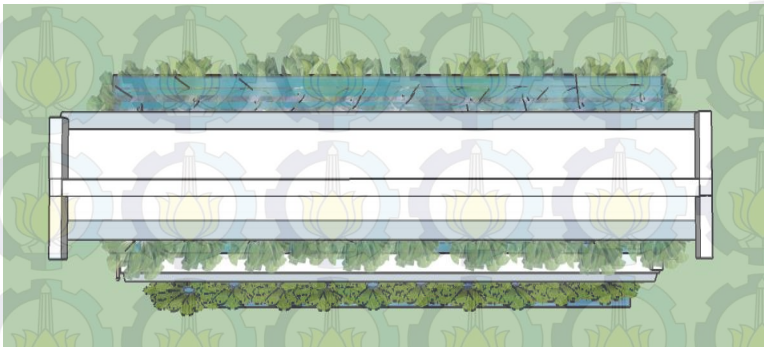
Gambar 3.15 Perancangan Panel Bagian Dalam



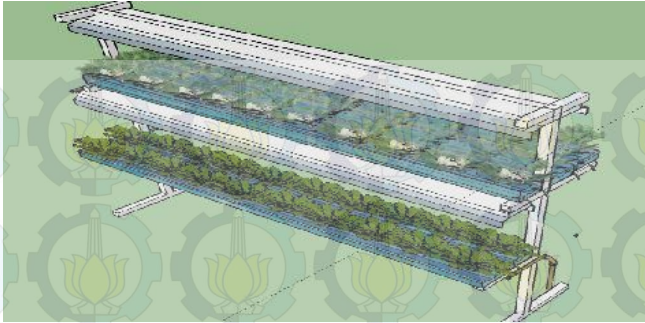
Gambar 3.16 Tampilan Panel

3.13 Perancangan *Hardware Plant*

Perancangan pembuatan *hardware plant* ini dibagi menjadi 2 perangkat keras. Pertama prototipe perkebunan hidroponik dengan sistem NFT (*Nutrient Film Technique*) dan yang kedua bak reservoir dengan panel. Prototipe perkebunan hidroponik ini dirancang menyerupai perkebunan mini hidroponik. Pembuatan prototipe ini bertujuan agar nantinya sistem ini dapat direalisasikan pada perkebunan Surabaya hidroponik. Sistem hidroponik yang digunakan pada prototipe ini menggunakan sistem hidroponik NFT (*Nutrient Film Technique*). Sistem hidroponik NFT adalah suatu metode hidroponik yang meletakkan akar tanaman pada aliran air yang dangkal yang menyerupai film. Dikarenakan hal ini maka sistem ini dikenal dengan sistem hidroponik NFT. Air tersebut tersirkulasi dan mengandung nutrisi yang dibutuhkan oleh tanaman sehingga perakaran dapat tumbuh dan berkembang. Prototipe ini dirancang seperti sistem hidroponik NFT dengan menggunakan talang bersusun, bak reservoir, kerangka bak reservoir dan panel, instalasi pipa pengisian, dan pipa pembuangan dari masing – masing pipa hidroponik hingga kembali ke bak reservoir. Dalam sistem otomatisasinya terdapat pompa nutrisi A dan pompa nutrisi B. Serta untuk pengganti cahaya ketika sistem ini ditambahkan LED *growlight*. Untuk lebih jelasnya perhatikan Gambar 3.17, gambar tersebut merupakan rancangan desain prototipe hidroponik yang terlihat dari atas. Dan Gambar 3.18 desain prototipe tampak dari depan.



Gambar 3.17 Desain Prototipe Hidroponik Tampak Atas



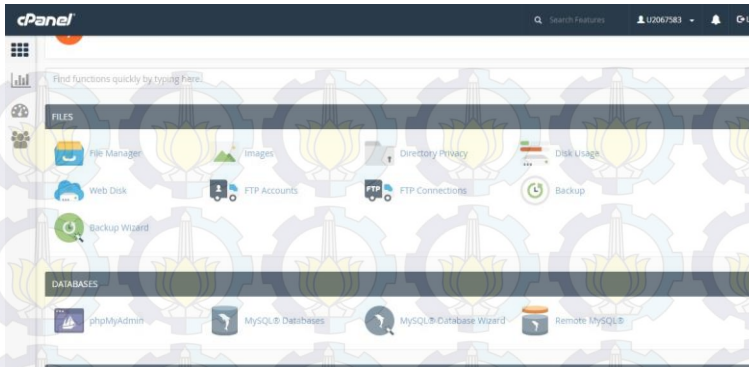
Gambar 3.18 Desain Prototipe Hidroponik Tampak Depan

3.14 Perancangan Web Hosting

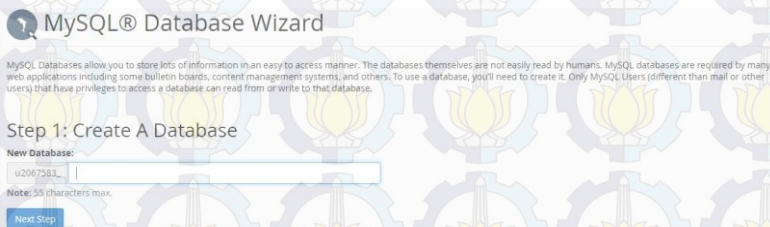
Dalam pembuatan *web hosting*, kami membeli pada salah satu reseller penyewaan *hosting*. Dalam menyiapkan *web hosting* harus mempunyai domain dan dilengkapi dengan *hosting*. Kami membeli di niagahoster.co.id dengan beberapa kelebihan yaitu waktu akses tercepat menggunakan cPanel dan server *web hosting* iix terbaik di Indonesia, garansi *uptime* 99.9% dengan *bandwidth unlimited*, domain untuk selamanya, dan dengan *disk space* yang *unlimited*. Dan juga mendukung beberapa fitur seperti Database MySQL, phpMyAdmin, Remote MySQL dan lain – lain. Di [niagahoster](http://niagahoster.com) kami membeli domain dengan nama androbos.com dengan sewa *hosting* dengan jangka waktu aktif selama 1 tahun.

3.15 Perancangan Database

Dalam menyiapkan database server terlebih dahulu membuat database MySQL terlebih dahulu di dalam cPanel. MySQL Database memungkinkan menyimpan dan mengakses informasi dalam jumlah besar dengan mudah. Fasilitas ini dibutuhkan oleh banyak aplikasi berbasis web, misalnya bulletin board, *Content Management System* (CMS), dan lain – lain. Hanya pengguna yang memiliki *privilege* atau kewenangan khusus yang dapat mengakses suatu database.



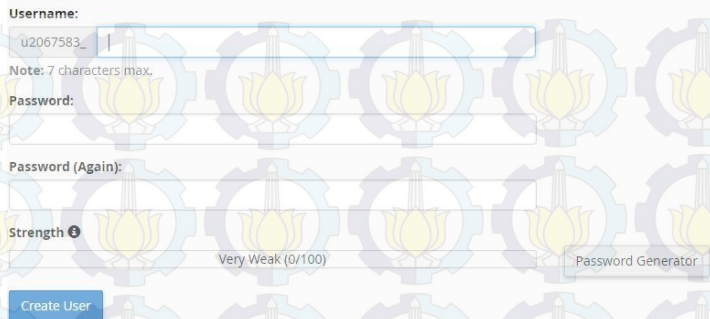
Gambar 3.19 Tampilan CPanel Secara *Cloud*



Gambar 3.20 Halaman *MySQL Database Wizard*

Cara untuk membuat database yaitu dengan menekan *MySQL Database Wizard* pada cPanel bagian databases seperti terlihat pada Gambar 3.19. lalu setelah menekan *MySQL Database Wizard* maka akan muncul halaman seperti pada Gambar 3.20 untuk membuat nama dari database yang nanti di inginkan. Namun dalam menentukan nama database harus diawali dengan *u2067583_* seperti terlihat pada Gambar 3.20. Disini kami membuat database dengan nama *u2067583_sim900* yang nantinya sebagai tempat menyimpan data yang dikirim oleh modul sim 900. Setelah selesai menyelesaikan step 1, selanjutnya yaitu step 2 membuat database *users* Seperti terlihat pada Gambar 3.21. setelah membuat nama dari database selanjutnya membuat *username* dan *password* yang digunakan untuk mengakses database yang telah dibuat.

Step 2: Create Database Users:




Username:

Note: 7 characters max.

Password:

Password (Again):

Strength  Very Weak (0/100)

[Password Generator](#)

[Create User](#)

Gambar 3.21 Tampilan Membuat Database *Users*



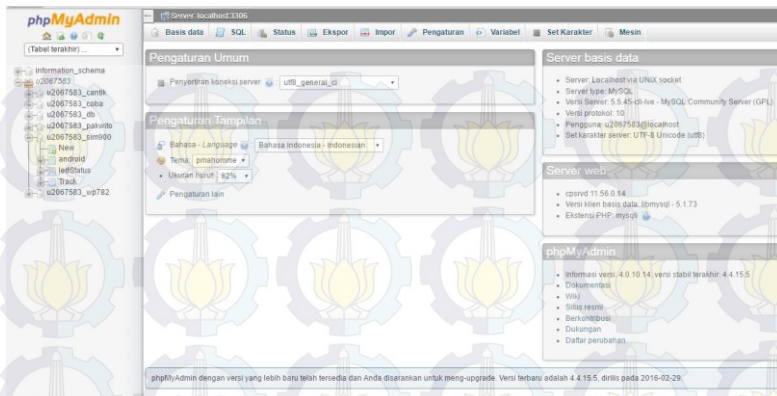
☒ ALL PRIVILEGES

<input checked="" type="checkbox"/> ALTER	<input checked="" type="checkbox"/> ALTER ROUTINE
<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> CREATE ROUTINE
<input checked="" type="checkbox"/> CREATE TEMPORARY TABLES	<input checked="" type="checkbox"/> CREATE VIEW
<input checked="" type="checkbox"/> DELETE	<input checked="" type="checkbox"/> DROP
<input checked="" type="checkbox"/> EVENT	<input checked="" type="checkbox"/> EXECUTE
<input checked="" type="checkbox"/> INDEX	<input checked="" type="checkbox"/> INSERT
<input checked="" type="checkbox"/> LOCK TABLES	<input checked="" type="checkbox"/> REFERENCES
<input checked="" type="checkbox"/> SELECT	<input checked="" type="checkbox"/> SHOW VIEW
<input checked="" type="checkbox"/> TRIGGER	<input checked="" type="checkbox"/> UPDATE

[Next Step](#)

Gambar 3.22 Tampilan Step Tiga Dalam Membuat Database

Setelah menentukan nama *username* dan *password* selanjutnya tekan tombol *Create User* jika telah selesai. Di langkah tiga akan muncul halaman seperti pada Gambar 3.22 yang diperkenalkannya memilih *privilege* atau kewenangan yang yang dapat dilakukan oleh *user* tertentu. *User* tersebut dapat menambah database, menghapus, dll. Step tiga ini sekaligus mengakhiri pembuatan database MySQL. Selanjutnya yaitu membuka phpMyAdmin pada cPanel untuk melihat database yang telah dibuat. Seperti pada Gambar 3.23.



Gambar 3.23 Tampilan *Php My Admin*

Server: localhost:3306 » Basis data: u2067583_sim900 » Tabel: android							
Jelajahi Struktur SQL Cari Tambahkan Ekspor Impor							
#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Ekstra	
<input type="checkbox"/>	1 id	int(100)		Tidak	Tidak ada	AUTO_INCREMENT	
<input type="checkbox"/>	2 hari	varchar(100) latin1_swedish_ci		Tidak	Tidak ada		
<input type="checkbox"/>	3 jam	varchar(100) latin1_swedish_ci		Tidak	Tidak ada		
<input type="checkbox"/>	4 tanggal	varchar(100) latin1_swedish_ci		Tidak	Tidak ada		
<input type="checkbox"/>	5 dataEC	int(100)		Tidak	Tidak ada		
<input type="checkbox"/>	6 dataLDR	int(100)		Tidak	Tidak ada		
<input type="checkbox"/>	7 dataPompa	int(11)		Tidak	Tidak ada		

Gambar 3.24 Struktur Tabel

Langkah selanjutnya yaitu membuat tabel yang nantinya diisi oleh data yang di kirim dari modul sim 900 berupa data sensor EC (*electrical conductivity*), hari , jam, tanggal, data dari LDR untuk menentukan hari sedang cerah atau mendung, dan data pompa. Dalam database u2067583_sim900 yang telah dibuat selanjutnya membuat tabel bernamn android dengan struktur tabel seperti pada Gambar 3.24. Nama id pada Gambar 3.24 berfungsi sebagai nomor setiap baris dengan jenis INT atau integer. Hari , jam, tanggal berjenis varchar atau karakter sedangkan dataEC, dataLDR, dan dataPompa bertipe integer.


```

1 <?php
2 mysql_connect("localhost", "u2067583_sim900", "sim900") or die(mysql_error());
3 mysql_select_db("u2067583_sim900") or die(mysql_error());
4
5 if( $_GET["hari"]|| $_GET["jam"]|| $_GET["tanggal"]|| $_GET["dataEC"]|| $_GET["dataLDR"]|| $_GET["dataPompa"]) {
6     $insert = "INSERT INTO android (hari, jam, tanggal, dataEC, dataLDR, dataPompa) VALUES
7     ('".$_GET['hari'].",". $_GET['jam'].",". $_GET['tanggal'].",". $_GET['dataEC'].",".
8     $_GET['dataLDR'].",". $_GET['dataPompa'].")";
9     if($insert){
10         echo "sukses ".$_GET['hari']. " " . $_GET['jam']. " " . $_GET['tanggal']. "
11         " . $_GET['dataEC']. " " . $_GET['dataLDR']. " " . $_GET['dataPompa']. " " . "<br />";
12     }
13     $add_member = mysql_query($insert);
14 }
15 }
16 >?

```

Gambar 3.25 Source Code Php Untuk Mengkoneksikan ke Database

http://yourphpserver/add_data.php?variable1=value1&variable2=value2&variable3=value3...

Gambar 3.26 Contoh Tampilan Url

<http://androbos.com/sim900/sim900/fix.php?hari=2&jam=10:23&tanggal=7/5/2016&dataEC=424&dataLDR=123&dataPompa=54>

Gambar 3.27 Tampilan Url

Untuk mengkoneksikan data yang dikirim dari modul sim 900 ke database yaitu dengan cara membuat URL seperti pada Gambar 3.26. dimana *value1*, *value2*, *value3* dan seterusnya merupakan data dari modul sim 900 dan akan masuk ke tabel dengan nama *variable1*, *variable2*, *variable3* dan seterusnya pada Gambar 3.26. untuk itu *source code* yang dibuat seperti pada Gambar 3.25 dengan penjelasan pada baris 1 dan 2 merupakan kode untuk mengkoneksikan ke database. Lalu pada baris 3 sampai 15 merupakan kode untuk memasukan data ke database. Pada baris 6 sampai 8 merupakan kode untuk mengkoneksikan data ke tabel android dan data dimasukan ke kolom hari, jam, tanggal, dataEC, dataLDR, dan dataPompa. *Source code* pada Gambar 3.25 akan menampilkan url seperti pada Gambar 3.27 dimana androbos.com merupakan domain yang telah dibeli di niagahoster. Format url Gambar 3.27 adalah format yang akan dibuat untuk mengirim data dari modul sim 900. Dari data yang telah berada pada database selanjutnya akan ditampilkan pada aplikasi *mobile* yaitu sistem operasi android. Untuk dapat ditampilkan di aplikasi android data tersebut diubah ke dalam

bentuk JSON (*Javascript Object Notation*) terlebih dahulu agar data tersebut dapat di *parsing* atau diterima oleh aplikasi android

3.16 Implementasi API dan JSON Parsing

Application Program Interface (API) bertujuan untuk menghubungkan aplikasi *mobile* dengan *database* yang ada di dalam *server*. API ini nantinya untuk untuk memenuhi seluruh jenis fungsi yang dibutuhkan di dalam aplikasi. Seperti fungsi tampil *item*, pembacaan EC dan lain sebagainya API ini pada akhirnya akan di *hosting* secara *online* sehingga dapat diakses secara *online*. API berisi Query `SELECT*FROM` seperti pada Gambar 3.28

Source code Gambar 3.28 berbasis PHP, berisikan Query `SELECT`, dan jika data ditemukan, data ditampilkan ke dalam bentuk JSON (`"json_encode"`). Data yang diubah ke dalam bentuk JSON adalah data yang dimulai dari yang terbaru ke data yang lama karena pada baris 4 terdapat fungsi `"DESC"` yang berarti data dari database di urutkan dari data yang paling bawah sampai paling atas. Di baris ke 5 ada fungsi `mysql_query` yang digunakan untuk mengirim query ke database MySQL. Fungsi `die()` digunakan untuk menampilkan pesan yang berada di dalam tanda kurung sekaligus menghentikan jalannya program. Dan `mysql_error()` digunakan untuk menghasilkan pesan error dari operasi `mysql`. Pada baris ke 7 terdapat fungsi `mysql_fetch_assoc()` yang berfungsi untuk menguraikan data yang diambil dari tabel database MySQL menjadi PHP array dengan indexnya adalah sesuai nama kolom tabel yang akan diambil datanya.

```
1 <?php
2 mysql_connect("localhost", "u2067583_sim900", "sim900") or die(mysql_error());
3 mysql_select_db("u2067583_sim900") or die(mysql_error());
4 $sql = "SELECT id, hari, jam, tanggal, dataEC, dataLDR, dataPompa FROM android ORDER BY id DESC";
5 $result = mysql_query($sql) or die ("Query error: " . mysql_error());
6 $arr = array();
7 while ($row = mysql_fetch_assoc($result)) {
8     $temp = array(
9         "hari" => $row["hari"],
10        "jam" => $row["jam"],
11        "tanggal" => $row["tanggal"],
12        "dataEC" => $row["dataEC"],
13        "dataEC" => $row["dataLDR"],
14        "dataPompa" => $row["dataPompa"], );
15
16    array_push($arr, $temp);
17 }
18 $data = json_encode($arr);
19 echo "{\n\data": " . $data . "\n}";
20 ?>
```

Gambar 3.28 *Source Code API*

Lalu dari Gambar 3.28 data yang diambil adalah data dari database berupa data hari, jam, tanggal, dataEC, dataLDR, dataPompa yang nantinya akan ditampilkan di dalam aplikasi *mobile*. Baris yang menunjukkan data yang akan diambil adalah dari baris 7 sampai baris ke 14. Lalu pada baris ke 16 ada fungsi `array_push` yang berfungsi untuk menambah satu atau lebih elemen baru di akhir array dengan keterangan `$arr` merupakan input array dan `$temp` merupakan nilai yang disisipkan di akhir array. Jika program dijalankan maka akan tampil *JSON code* seperti pada Gambar 3.29 yang merupakan bentuk *JSON* dari data hasil Query dari API yang dibuat. Selanjutnya, *JSON code* ini akan dilanjutkan ke dalam aplikasi *mobile* untuk diambil datanya dan ditampilkan.

```
{
  "data": [
    {
      "hari": "2",
      "jam": "10:23",
      "tanggal": "7/5/2016",
      "dataEC": "123",
      "dataPompa": "54"
    },
    {
      "hari": "2",
      "jam": "10:23",
      "tanggal": "7/5/2016",
      "dataEC": "123",
      "dataPompa": "54"
    },
    {
      "hari": "1",
      "jam": "23:20",
      "tanggal": "5/5/2016",
      "dataEC": "190",
      "dataPompa": "3"
    }
  ]
}
```

Gambar 3.29 Tampilan *JSON Code*

APLIKASI
MOBILE



API



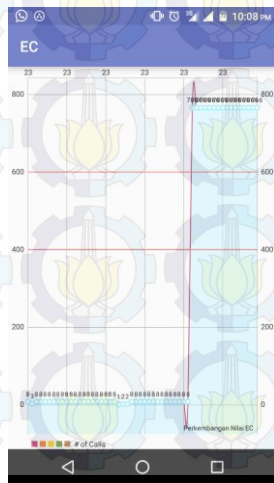
DATABASE

Gambar 3.30 Arsitektur Aplikasi

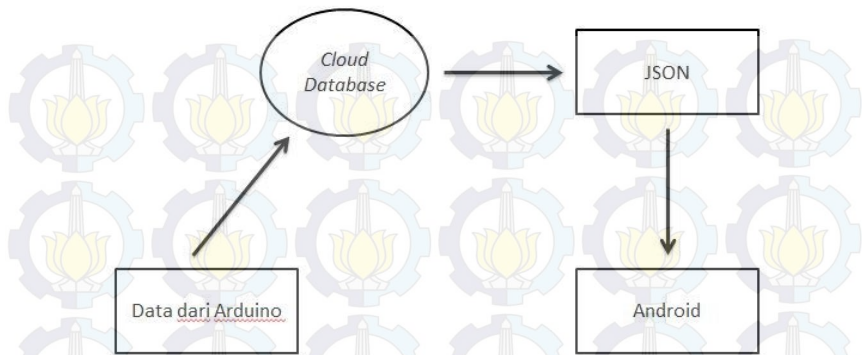
3.17 Perancangan Aplikasi *Mobile*

Pada perancangan dalam aplikasi *mobile* lebih di khususkan pada aplikasi sistem operasi android. Berdasarkan Gambar 3.30 mengenai arsitektur aplikasi, secara umum gambaran sistem adalah aplikasi menampilkan data *website* dengan menggunakan API (*Application Program Interface*) yang datanya diubah dalam bentuk JSON (*Javascript Object Notation*) agar data dapat di olah oleh aplikasi sistem operasi android. data – data tersebut berasal dari database server secara *cloud* atau online sehingga dapat diakses oleh siapa saja yang memiliki aplikasi androidnya dan tersambung dengan internet. Tampilan aplikasi android seperti pada Gambar 3.31 yang menampilkan data waktu, hari tanaman, dan nilai EC (*Electrical Conductivity*). Pembuatan aplikasi android menggunakan software Android Studio karena lebih cepat dalam pembuatan aplikasinya dan selalu mendapatkan *update* atau pemberitahuan terbaru mengenai pengembangan aplikasi android dan *open – source* alias *free* atau bebas digunakan.

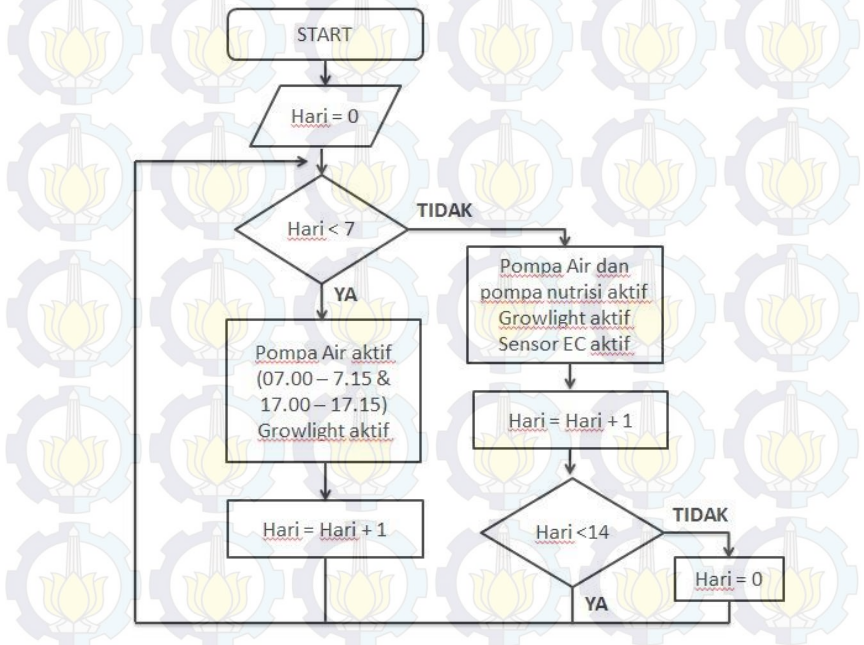
Lalu pada Gambar 3.32 merupakan skema aliran data dari data yang dikirim dari arduino melalui modul SIM 900 ke *cloud database* yang nantinya akan ditampilkan pada layar aplikasi android. Pada saat sistem dijalankan, program akan mendeteksi sekarang hari. Bila belum lebih hari ketujuh maka sensor EC dan pompa nutrisi belum dijalankan karena akan dijalankan setelah hari ke tujuh. Seperti pada Gambar 3.33.



Gambar 3.31 Tampilan Aplikasi Android



Gambar 3.32 Skema Perangkat Lunak



Gambar 3.33 Diagram Alur Program



BAB IV

PENGUKURAN DAN ANALISA

Untuk mengetahui apakah tujuan – tujuan dari pembuatan alat ini telah sesuai dengan yang diharapkan atau tidak, maka dilakukan pengujian dan analisa terhadap alat yang telah dibuat.

4.1 Cara Kerja Alat

Cara kerja dari alat yang telah dibuat yaitu pada saat hari pertama sampai hari ke tujuh menyalakan pompa air setiap pukul 07.00 – 07.15 WIB dan pukul 17.00 – 17.15 WIB. Lalu pada hari ke 8 sampai hari ke 14 pompa nutrisi AB mix dan sensor EC (*Electrical Conductivity*) baru aktif untuk mendeteksi kadar EC pada air dan pompa air akan terus mengalir. Saat hari 8 sampai hari ke 14 sensor EC mendeteksi EC pada air yang mengalir dengan beberapa aturan yaitu bila kadar EC kurang dari 900 uS/cm, pompa nutrisi aktif. Jika kadar EC antara 900 – 1200 uS/cm pompa nutrisi mati. Jika kadar EC pada air lebih dari 1200 uS/cm alarm akan aktif. Lalu sensor LDR akan aktif mendeteksi cahaya mulai hari pertama sampai hari terakhir yaitu hari ke 14 dari pukul 06.00 – 18.00 WIB. Sensor LDR mendeteksi keadaan cuaca sedang mendung ataupun cerah mengeluarkan nilai PWM untuk mengatur intensitas cahaya dari *growlight*. Semua waktu dilakukan oleh RTC (Real Time Clock).

4.2 Pengujian Hardware (Perangkat Keras)

Pada pengujian ini menguji *Power Supply* yang dibuat, Pengujian Rangkaian Driver Relay, Pengujian Lampu LED *Growlight*, pengujian pompa air, pengujian pompa nutrisi, pengujian sensor EC (*Electrical Conductivity*), pengujian modul SIM 900, dan Pengujian RTC (*Real Time Clock*).

4.2.1 Pengujian Power Supply

Pengujian *power supply* 5, 9, dan 12 volt DC bertujuan untuk mengetahui drop tegangan serta arus ketika *power supply* sebelum disambungkan ke beban, dan setelah disambungkan ke beban.



Gambar 4.1 *Power Supply*

Tabel 4.1 Hasil Pengukuran *Power Supply*

Power Supply	Output (volt)			Error (%)
	1	2	3	
+5 V	5.06	5.06	5.06	1.18
+9	8.9	8.9	8.9	1.12
+12 V	11.85	11.85	11.85	1.26

Pengujian dilakukan dengan mengukur keluaran dengan voltmeter, Cara pengujian *power supply* dilakukan dengan mengukur tegangan output power supply. Pengujian dilakukan dengan mengukur keluaran voltmeter kemudian dihitung presentasi error. Dengan cara:

% Error = (Nilai acuan – nilai pengukuran) / Nilai acuan x 100%
kemudian dihitung presentasi error.

% Error = (Nilai pengukuran – nilai acuan) / Nilai pengukuran x 100 %

% Error 5 volt = $(5.06 - 5) / 5.06 \times 100\% = 1.18\%$

% Error 9 volt = $(8.9 - 9) / 8.9 \times 100\% = 1.12\%$

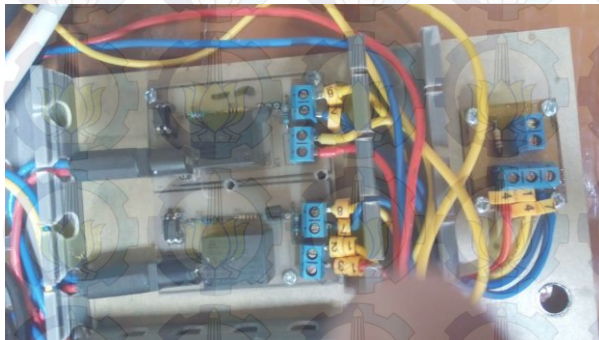
% Error 12 volt = $(11.85 - 12) / 11.85 \times 100\% = 1.26\%$

Dari hasil tegangan diatas dapat diketahui bahwa pada *supply* +5 volt, output tegangan yang terukur sampai ketitik kestabilan mencapai +5.06 volt. Pada *supply* +9 volt, output tegangan berhenti di 8.9 volt. Dan pada *supply* +12 volt, output tegangannya berhenti pada 11.85 volt.

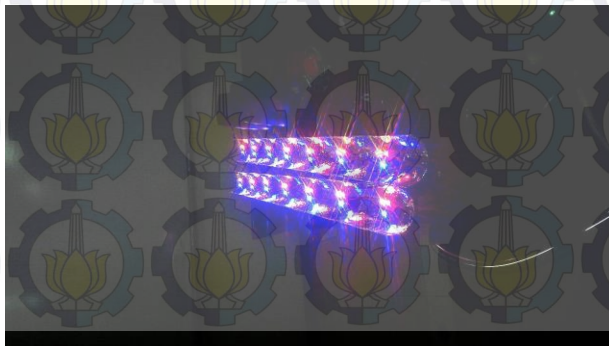
Maka dari itu, *power supply* ini dapat digunakan dalam sistem ini untuk men – supply komponen atau instrumen dalam sistem ini.

4.2.2 Pengujian Rangkaian Driver Relay

Pengujian ini dilakukan agar dapat mengetahui relay dapat bekerja sesuai sistem agar pompa nutrisi, pengaduk, grow light, maupun pompa air dapat terkontrol dengan baik. Metode pengujian rangkaian *relay* dilakukan dengan menerapkan program sederhana pada mikrokontroler arduino. Status *ON* pada relay diperoleh jika basis pada transistor C829 diberi input *logic* 1 dari mikrokontroler atau diberi tegangan +5 volt yang telah memberikan logic aktif pada optocoupler PC817. Sedangkan apabila basis pada transistor C829 diberi *input* 0 dari mikrokontroler arduino atau diberi tegangan 0 volt, maka *relay* akan berganti status menjadi *OFF*.



Gambar 4.2 Pengujian Driver Relay



Gambar 4.3 Pengujian Growlight

Pada alat ini, relay digunakan untuk mengaktifkan pompa nutrisi dan pompa air.

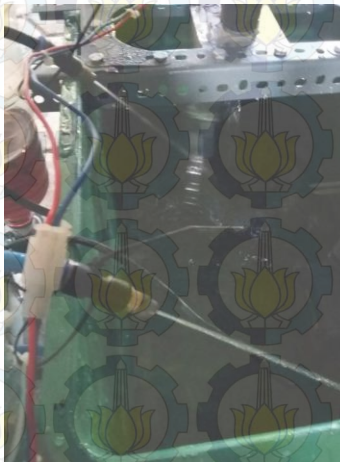
4.2.3 Pengujian Lampu LED *Growlight*

Pengujian ini dilakukan agar dapat mengetahui aktivasi LED *growlight*. LED dengan perbandingan 3:1 untuk LED warna merah dan biru. Metode pengujian dilakukan dengan cara menyuplai semua led dengan tegangan 5 volt. Dan hasilnya semua LED dapat bersinar dan menghasilkan spektrum warna ungu.

Pada Gambar 4.3, terlihat LED *growlight* dapat menyala dengan catu daya 5 volt dari pinout arduino untuk mengaktifkan *driver relay* untuk menyuplai tegangan ke *growlight*. Maka dari itu *growlight* dapat berfungsi dengan baik.

4.2.4 Pengujian Pompa Nutrisi

Pengujian pompa nutrisi dengan cara mengecek nilai EC (*Electrical Conductivity*). Jika nilai EC kurang dari 400 ppm maka pompa nutrisi aktif. Jika nilai EC diantara 400 dan 600 maka pompa nutrisi tidak aktif. Dan jika nilai EC lebih dari 600 maka akan ada buzzer yang bunyi menandakan nilai EC sudah melebihi batas yang dibutuhkan.



Gambar 4.4 Pengujian Pompa Nutrisi

4.2.5 Pengujian Sensor EC (*Electrical Conductivity*)

Pada pengujian sensor EC (*Electrical Conductivity*) pembacaan nilai ADC yang dibaca oleh sensor YL – 69 dibandingkan dengan nilai sensor EC yang sebenarnya. Didapatkan data seperti pada Tabel 4.2.

Pada Tabel 4.2 terlihat bahwa pembacaan nilai EC yang semakin akan semakin rendah terhadap pembacaan sensor YL – 69. Dan pada nilai EC yang semakin tinggi, nilai ADC dari sensor YL – 69 akan semakin menurun dengan penurunan yang semakin kecil. Ini dikarenakan pada pembacaan sensor YL - 69 ada suatu titik dimana nilai pembacaan berada di titik maksimal pembacaan.

4.2.6 Pengujian Modul SIM 900

Pada pengujian modul SIM 900 data yang dikirim akan ditampilkan pada url yang telah dibuat yaitu url pada Gambar 4.5. Semua data yang dikirim oleh modul SIM 900 dapat dilihat di url pada Gambar 4.5. Tampilan dari url tersebut adalah berupa JSON (*Javascript Object Notation*) karena dari url tersebut yang isinya berupa JSON akan ditampilkan ke dalam aplikasi *mobile* khususnya aplikasi android.

Tabel 4.2 Tabel Pengukuran Nilai EC

Nilai EC	Nilai pembacaan sensor YL – 69
513	812
677	873
795	875
885	876
999	900
1140	902
1271	909
1447	1375
1590	1576
1950	1765

 androbos.com/sim900/sim900/json.php

Gambar 4.5 Url Menampilkan Data Oleh Modul SIM 900

4.2.7 Pengujian RTC (*Real Time Clock*)

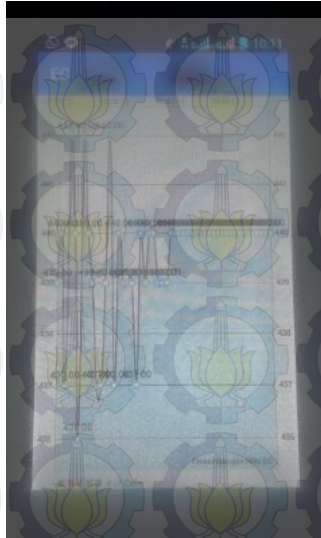
[illegible]

Gambar 4.6 Tampilan JSON



Gambar 4.7 Pengujian RTC

4.3 Pengujian *Software* (Perangkat Lunak)



Gambar 4.8 Pengujian Tampilan Aplikasi Android

Dalam pengujian perangkat lunak pada sistem ini yaitu melihat perubahan nilai yang diterima oleh aplikasi android apabila data yang dikirim oleh modul SIM 900 ke *server* dan data tersebut dijadikan ke bentuk JSON (*Javascript Object Notation*) dapat ditampilkan di aplikasi android. Yang ditampilkan dalam aplikasi android yaitu nilai EC (*Electrical Conductivity*), hari tanaman, dan waktu seperti pada Gambar 4.8.

4.4 Pengujian Keseluruhan

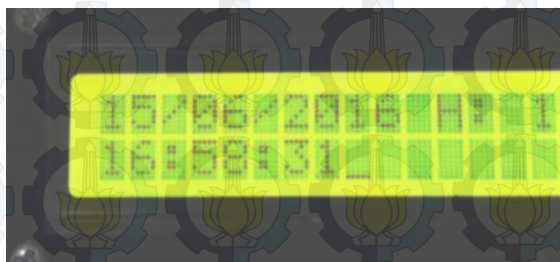
Pengujian sistem ini dilakukan dengan dua tahapan yaitu pengujian saat hari kurang dari tujuh dan hari lebih dari tujuh. Saat hari kurang dari tujuh tahapan yang terjadi adalah pompa air akan aktif pada pukul 07.00 – 07.15 WIB dan 17.00 – 17.15 WIB. Lalu saat hari lebih dari tujuh tahapan yang terjadi adalah pompa air akan aktif terus menerus sedangkan pompa nutrisi dan sensor EC (*Electrical Conductivity*) mulai aktif.

4.4.1 Hari Kurang Dari Tujuh

Saat hari kurang dari tujuh bagian sistem yang aktif adalah pompa air dan *growlight*. Pompa air akan aktif pada pukul 07.00 – 07.15 dan 17.00 – 17.15. Sedangkan untuk *growlight* akan aktif dengan ketentuan waktu menunjukkan pukul 07.00 – 17.00 dan dipengaruhi oleh sensor LDR. Jadi, ketika LDR mendeteksi kondisi lingkungan kekurangan cahaya pada jam 07.00 – 17.00 maka *growlight* akan aktif. Berikut adalah simulasi sistem pada hari kurang dari tujuh. Pada Gambar 4.9 menunjukkan pompa air belum berjalan dikarenakan waktu belum menunjukkan pukul 07.00 WIB dan 17.00 WIB seperti di tunjukkan pada Gambar 4.10 yang menunjukkan pukul 16.58 WIB.

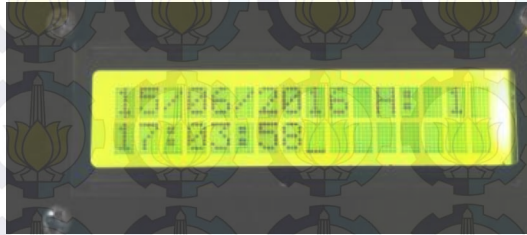


Gambar 4.9 Pompa Air Belum Aktif



Gambar 4.10 Waktu Pada Sistem

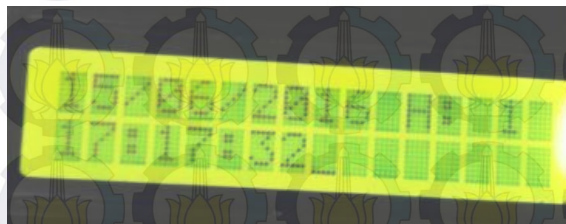
Pada Gambar 4.11 waktu sudah menunjukkan pukul 17.03 maka pompa air akan aktif seperti di tunjukkan pada Gambar 4.12. Kemudian pada Gambar 4.13 menunjukkan pukul 17.17 maka pompa air akan berhenti seperti di tunjukkan pada Gambar 4.14.



Gambar 4.11 Penunjuk Waktu



Gambar 4.12 Pompa Air Aktif



Gambar 4.13 Waktu Menunjukkan Pukul 17.17



Gambar 4.14 Pompa Air Berhenti



Gambar 4.15 Waktu Menunjukkan Pukul 07.02



Gambar 4.16 Pompa Air Aktif

Pada Gambar 4.15 waktu sudah menunjukkan pukul 07.02 maka pompa air akan aktif seperti di tunjukkan pada Gambar 4.16 .



Gambar 4.17 Waktu Menunjukkan Pukul 07.16



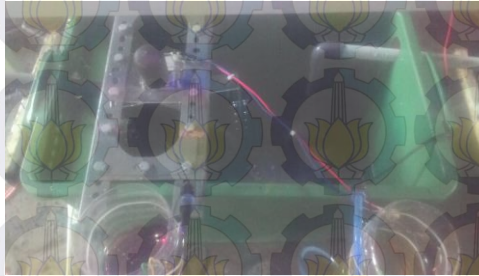
Gambar 4.18 Pompa Air Berhenti

Kemudian pada Gambar 4.17 menunjukkan pukul 07.16 maka pompa air akan berhenti seperti di tunjukkan pada Gambar 4.18.

4.4.2 Hari Lebih Dari Tujuh

Saat hari lebih dari tujuh bagian sistem yang aktif adalah pompa air, pompa nutrisi, *growlight* dan pengaduk. Pompa air dan pengaduk akan aktif terus menerus. *Growlight* akan aktif dengan ketentuan waktu menunjukkan pukul 07.00 – 17.00 dan dipengaruhi oleh sensor LDR. Sedangkan pompa nutrisi akan aktif ketika kadar EC (*Electrical Conductivity*) kurang dari 900 $\mu\text{S}/\text{cm}$. Ketika kadar nutrisi diantara 900

uS/cm maka pompa nutrisi tidak aktif. Pada Gambar 4.19 dan Gambar 4.20 menunjukkan pengaduk dan pompa air telah aktif karena RTC telah menunjukkan hari lebih dari tujuh seperti di tunjukkan pada Gambar 4.21.



Gambar 4.19 Pengaduk

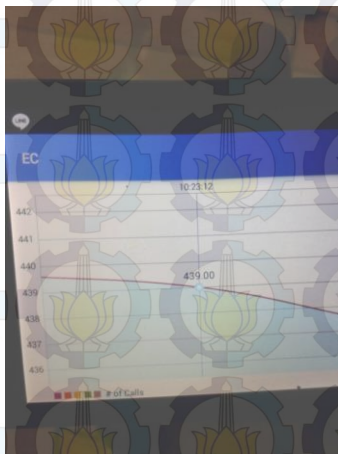


Gambar 4.20 Pompa Air Aktif



Gambar 4.21 Hari Menunjukkan Lebih Dari Tujuh

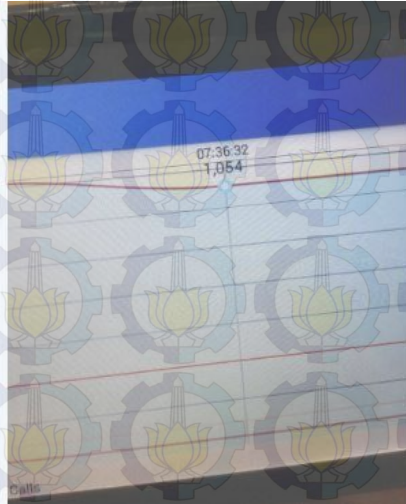
Gambar 4.23.



Gambar 4.22 Tampilan Kadar EC Pada Aplikasi Android



Pada Gambar 4.24 menunjukkan kadar nutrisi diantara 900 uS/cm – 1200 uS/cm maka pompa nutrisi tidak aktif seperti yang di tunjukkan pada Gambar 4.25.

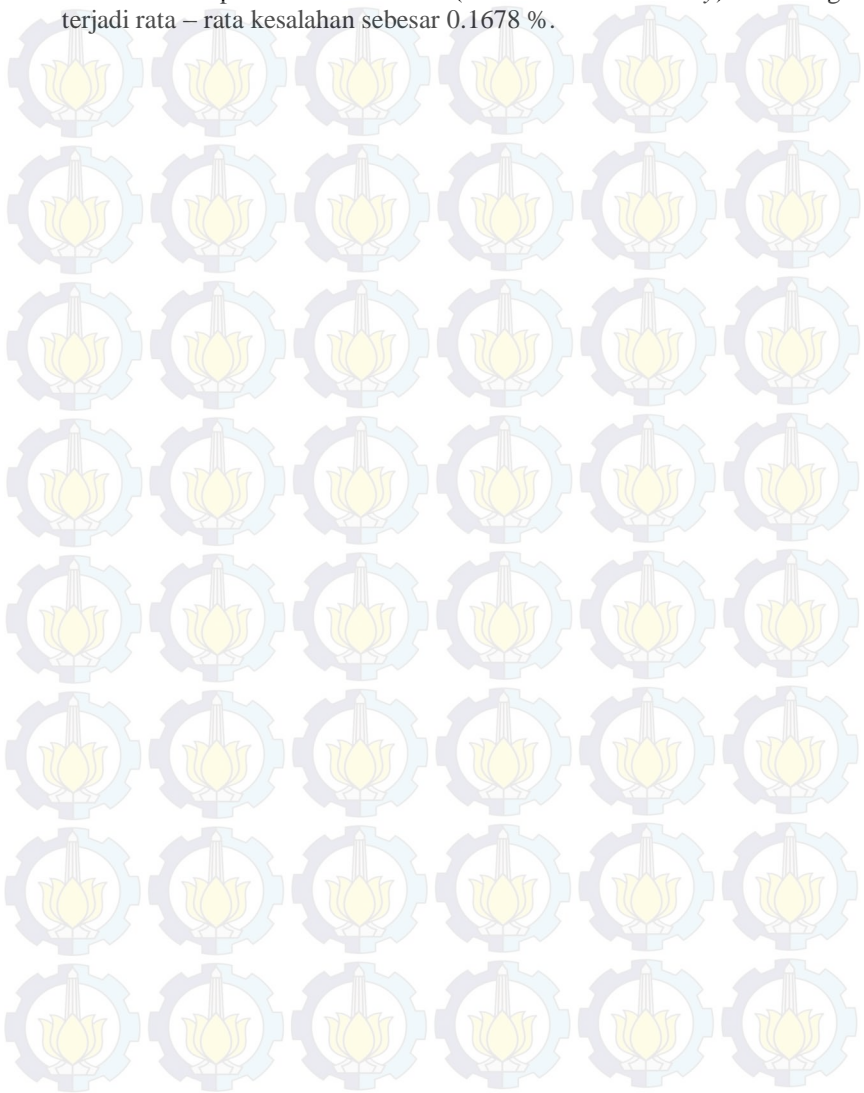


Gambar 4.24 Tampilan Kadar EC Pada Aplikasi Android



Gambar 4.25 Pompa Nutrisi Mati

Dari Siklus tersebut sistem berjalan dengan sesuai yang diharapkan namun dalam pembacaan nilai EC (*Electrical Conductivity*) terkadang terjadi rata – rata kesalahan sebesar 0.1678 %.



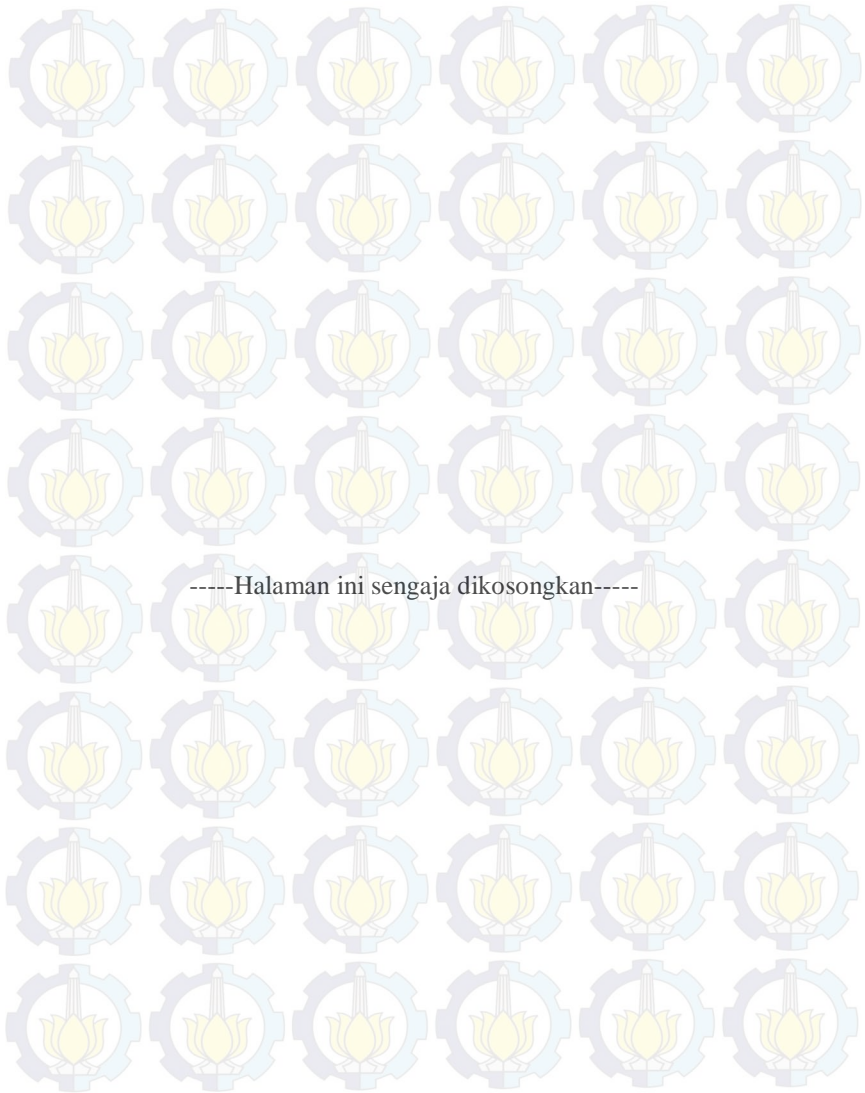


BAB V

PENUTUP

Dari hasil pengujian sistem kontrol hasil desain dengan alat yang dibuat, dapat diambil kesimpulan bahwa nilai rata – rata kesalahan dari pengukuran EC (*Electrical Conductivity*) sebesar 0.1678 % dengan pemberian nutrisi pada hari ke 8 sampai 14 saat selada memiliki 3 atau 4 helai daun. Lalu , proses pengiriman data EC (*Electrical Conductivity*) melalui modul sim 900 membutuhkan waktu 60 detik saat hari menunjukkan lebih dari 7 hari.

Saran untuk penelitian selanjutnya adalah untuk lebih memerhatikan nilai EC dan penerangan dengan *Growlight* di tempat tertutup.



DAFTAR PUSTAKA

- [1] Mas'ud, Hidayati. , "Sistem Hidroponik Dengan Nutrisi dan Media Tanam Berbeda Terhadap Pertumbuhan dan Hasil Selada ", *Media Litbang Sulteng* 2 (2):131 - 136, Sulteng, Desember, 2009.
- [2] Zuhaida, Laila., Ambarwati, Erlina., dan Sulistyaningsih, Endang. , "Pertumbuhan dan Hasil Selada (*Lactuca sativa* L.) Hidroponik Diperkaya Fe", *Tugas Akhir*, Fakultas Pertanian Universitas Gajah Mada, Yogyakarta, 2014.
- [3] Siswadi., dan Yuwono, Teguh. , "Pengaruh Macam Media Terhadap Pertumbuhan dan Hasil Selada (*Lactuca sativa* L) Hidroponik", *Jurnal Agronomika Vol. 09 No. 03*, Surakarta, 2015.
- [4] Kustanti, Ika., Muslim, M. A., Yudaningsih, Erni. , "Pengendalian Kadar Keasaman (pH) Pada Sistem Hidroponik Stroberi Menggunakan Kontroler PID Berbasis Arduino Uno", *Jurnal Ika Kustanti*, Surabaya, 2014.
- [5] Syariefa, Evi., dkk. , "Hidroponik Praktis", *PT. Trubus Swadaya*, Depok, 2015.
- [6] Christofer, Gerry., Sujaini, Herry., Irwansyah, M.Azhar., "Rancang Bangun Aplikasi *Early Warning* Dengan Pemanfaatan Pengukuran Suhu Ruangan Berbasis Arduino Mega 2560", *Tugas Akhir*, Program Studi Informatika Universitas Tanjungpura, Tanjungpura, 2014.
- [7] Ogata K. , "*Modern Control Engineering*", Prentice-Hall, New Jersey, Ch. 3, 1997.
- [8] Wibowo, Sapto., dan Asriyanti, Arum. , "Aplikasi Hidroponik NFT pada Budidaya Pakcoy (*Brassica rapa chinensis*)", *Jurnal Penelitian Pertanian Terapan Vol.13 (3)*:159-167, Banjarnegara, 2013.
- [9] Vrileuis, Adam. , "Pemantau Lalu Lintas dengan Sensor LDR Berbasis Mikrokontroler ATmega 16", *Jurnal Rekayasa*

Elektrika, 2013.

- [10] Bnagun, Wijayanto. , "Prototype Aplikasi Tumbuh Kembang Balita Berbasis Android untuk Kader Posyandu di Pedesaan", *Seminar Nasional Aplikasi Teknologi Informasi*, Yogyakarta, 2012.
- [11] Rachman, Isa., Subianto, Lilik., Suhardjito, Gaguk dan Indartono, Arie. , "Identifikasi Garis Stabilitas Melintang Kapal Melalui Percobaan Kemiringan Menggunakan DELPHI Berbasis Arduino", *Transmisi 16 (3)*:122, Surabaya, 2014.
- [12] Smyth, Neil. , "Android Studio Development Essentials – Secont Edition", *Ebook Frenzy*, Juni, 2015.
- [13] Sulistiono, Adi Achmad,. Sumardi., dan Agus, Munawar. , "Perancangan Sistem Pengendali Pada *Prototype* Rumah Jamur Menggunakan PLC Omron CPM1A", *TRANSIENT Vol.4 no.3*, 2015.
- [14] Yenny, Ervina. , "Rancang Bangun Sistem Penyiraman Sayur Sawi (*Brassica chinensi L.*) Menggunakan Sensor Kelembaban dan Sensor Intensitas Cahaya Berbasis *Fuzzy Logic*", *Tugas Akhir*, Universitas Jember, Jember, 2015.

LAMPIRAN A

A.1. Program Arduino

```
#include <Wire.h>
#include <RtcDS1307.h>
#include <SoftwareSerial.h>
#define pinPowerSIM900 9
#include <LiquidCrystal_I2C.h>
SoftwareSerial SIM900(0, 1); // configure software serial port
RtcDS1307 RTC;
LiquidCrystal_I2C lcd(0x27, 16, 2);
String datestring1= "";
String hourstring1="";
float nilaiEC,nilaiEC1,nilaiEC2;
float cons,h_x,h_x2,h_x3,x2,x3;
int sensorECPin = A0;
int sensorECBaca;
float nilaiECFinal;
byte pompaNutrisiIndikator = 0;
int pompaAirIndikator;
int pompaAirPin = 13;
int hari = 0;
byte hariIniJam = 6;
byte hariIniMenit = 1;
byte hariIniDetik = 1;
int pompaAir;
byte pompaAirJamAwalPagi = 7;
byte pompaAirMenitAwalPagi = 1;
byte pompaAirJamAkhirPagi = 7;
byte pompaAirMenitAkhirPagi = 16;
byte pompaAirJamAwalMalam = 17;
byte pompaAirMenitAwalMalam = 1;
byte pompaAirJamAkhirMalam = 17;
byte pompaAirMenitAkhirMalam = 16;
int LdrInputPin = A1 ;
int ldrOutputPin = 3 ;
byte ldrIndikator = 0;
byte ldrJamNyalaPagi = 6;
```

```

byte ldrMenitNyalapagi = 1;
byte ldrJamMatiMalam = 18;
byte ldrMenitMatiMalam = 1;
int pompaNutrisiPin1 = 5 ;
int pompaNutrisiPin2 = 2 ;
int pompaNutrisiNilai;
int growlight;
int bacaLdr ;
int bacaLdrKalibrasi;
int relaypin = 4;
int buzzerPin = 8;
int levelNormal = 7 ;
int levelMinNormal =6;
int pengadukPin = 12;
int pengaduk;
int adcEC;
String EC;
int pompanutrisiA = 2;
int pompaNutrisiB = 5;
int EC2;
int EC1;
int tambah;
int bacaLdrKalibrasi2;
void setup() {
  SIM900.begin(19200);
  SIM900power();
  lcd.begin();
  Serial.println(__DATE__);
  Serial.println(__TIME__);
  Wire.begin();
  RTC.Begin();
  RtcDateTime compiled = RtcDateTime(__DATE__,
__TIME__);
  printDateTimeKalibrasi(compiled);
  Serial.println();
  if (!RTC.IsDateTimeValid())
  {

```



```

    RTC.SetDateTime(compiled);
}

if (!RTC.GetIsRunning())
{
    RTC.SetIsRunning(true);
}

RtcDateTime now = RTC.GetDateTime();
if (now < compiled)
{
    RTC.SetDateTime(compiled);
}
else if (now > compiled)
{
    Serial.println("RTC is newer than compile time. (this is
expected)");
}
else if (now == compiled)
{
    Serial.println("RTC is the same as compile time! (not
expected but all is fine)");
}

RTC.SetSquareWavePin(DS1307SquareWaveOut_Low);
////////////////////
// put your setup code here, to run once:
pinMode(pompaAirPin,OUTPUT);
pinMode(pengadukPin,OUTPUT);
pinMode(relaypin,OUTPUT);
pinMode(ldrOutputPin,OUTPUT);
datestring1= "";
hourstring1="";
EC="";
tambah=0;
pinMode(pompanutrisiA,OUTPUT);
pinMode(pompaNutrisiB,OUTPUT);
}

```

```

void SIM900power()
{
    digitalWrite(9, HIGH);
    delay(2000);
    digitalWrite(9, LOW);
    delay(3000);
}

void SendSQL(){
    EC2=EC1;
    bacaLdrKalibrasi2=bacaLdrKalibrasi;
    RtcDateTime now = RTC.GetDateTime();
    SIM900.println("AT+SAPBR=3,1,\"Contype\", \"GPRS\");
    delay(500);
    SIM900.println("AT+SAPBR=3,1,\"APN\", \"internet\");
    delay(500);
    SIM900.println("AT+SAPBR=1,1");
    delay(500);
    SIM900.println("AT+SAPBR=2,1");
    delay(3000);
    SIM900.println("AT+HTTPIPINIT");
    delay(500);
    SIM900.println("AT+HTTTPARA=\"CID\",1");
    delay(500);

    SIM900.print("AT+HTTTPARA=\"URL\", \"http://androbos.com/sim900/sim900/fix.php?");
    //
    hari=8&jam=8&tanggal=5&dataEC=54&dataLDR=5&dataPompa=54")
    ;
    SIM900.print("hari=");
    SIM900.print(now.Day());
    //Serial.print("/");
    SIM900.print("&jam=");
    SIM900.print(now.Hour());
    SIM900.print(":");
    SIM900.print(now.Minute());
    SIM900.print(":");

```

```

SIM900.print(now.Second());
SIM900.print("&tanggal=");
SIM900.print(now.Day());
SIM900.print("-");
SIM900.print(now.Month());
SIM900.print("-");
SIM900.print(now.Year());
SIM900.print("&dataEC=");
SIM900.print(EC2);
SIM900.print("&dataLDR=");
SIM900.print(bacaLdrKalibrasi2);
SIM900.print("&dataPompa=5");
//bacaLdrKalibrasi
// nilaiIECFinal

SIM900.println("");
delay(3000);
SIM900.println("AT+HTTPACTION=0");
delay(15000);
SIM900.println("AT+HTTPTERM");
delay(500);
SIM900.println("AT+SAPBR=0,1");
delay(500);
}

void loop() {
// digitalWrite(pompanutrisiA,HIGH);
//digitalWrite(pompaNutrisiB,HIGH);
// put your main code here, to run repeatedly:
RtcDateTime now = RTC.GetDateTime();
ldrIndikator=1;
pompaAirIndikator=1;
startPompaAir();
pengaduk1();
printDateTime(now);

ldrAktif();
growlightAktif());

```

```

ECfinal();
lcd.clear();
    lcd.cursor();
    lcd.setCursor(0,0);
    lcd.print(datestring1);
    lcd.setCursor(0,1);
    //nilaiECFinal
    // lcd.noBlink();
    lcd.print(hourstring1);
    lcd.setCursor(9,1);
    lcd.print("EC:");
    lcd.print(EC);////diG
    Serial.print(datestring1);Serial.print("");
    Serial.print(hourstring1);Serial.print(" ");
    Serial.println(EC);
    if(now.Second()==byte(12)){

        tambah=5;
    }
    if (now.Second(>byte(12)){
        tambah=6;
    }

    if(tambah==5){
        tambah=6;
    }
    if(tambah==6){
        tambah=4;
    }
    //SendSQL();
    //SIM900power();
    }
    // lcd.noBlink();

}

void startPompaAir(){
    RtcDateTime now = RTC.GetDateTime();

    if (now.Hour()==pompaAirJamAwalPagi && now.Minute()
==pompaAirMenitAwalPagi)

```



```

{
    pompaAirIndikator = 1;
}
if(now.Hour() == pompaAirJamAkhirPagi && now.Minute() ==
pompaAirMenitAkhirPagi)
{
    pompaAirIndikator = 0;
}
/////
if (now.Hour()==pompaAirJamAwalMalam && now.Minute()
==pompaAirMenitAwalMalam)
{
    pompaAirIndikator = 1;
}
if(now.Hour() == pompaAirJamAkhirMalam && now.Minute()
== pompaAirMenitAkhirMalam)
{
    pompaAirIndikator = 0;
}

if(pompaAirIndikator == 1){
    pompaAir = HIGH;
    digitalWrite( pompaAirPin,pompaAir );
}
if(pompaAirIndikator == 0){
    pompaAir = LOW;
    digitalWrite( pompaAirPin,pompaAir );
}
}

void pengaduk1(){
    pengaduk=HIGH;
    digitalWrite(pengadukPin,pengaduk);
}

#define countofc(a) (sizeof(a) / sizeof(a[0]))

void printDateTimeKalibrasi(const RtcDateTime& dt)
{
    char datestringb[20];

```

```

    snprintf_P(datestringb,
        countofc(datestringb),
        PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
        dt.Month(),
        dt.Day(),
        dt.Year(),
        dt.Hour(),
        dt.Minute(),
        dt.Second() );
    Serial.println(datestringb);
}

#define countof(a) (sizeof(a) / sizeof(a[0]))
#define countofb(b) (sizeof(b) / sizeof(b[0]))
void printDateTime(const RtcDateTime& dt)
{
    char datestring[20];
    char hourstring[20];

    snprintf_P(datestring,
        countofc(datestring),
        PSTR("%02u/%02u/%04u "),
        dt.Month(),
        dt.Day(),
        dt.Year()
    );

    snprintf_P(hourstring,
        countofb(hourstring),
        PSTR("%02u:%02u:%02u"),
        dt.Hour(),
        dt.Minute(),
        dt.Second()
    );

    // Serial.print(datestring);
    // Serial.print(" ");

```

```

// Serial.print(hourstring);
//lcd.clear();
// lcd.cursor();
//lcd.setCursor(0,0);
//lcd.print(datestring);
// lcd.setCursor(0,1);
// lcd.noBlink();
//lcd.print(hourstring);
// lcd.noBlink();
datestring1=datestring;
hourstring1=hourstring;
delay(200);
}

```

```

void ldrAktif(){
    RtcDateTime now = RTC.GetDateTime();
    if(now.Hour()== ldrJamNyalaPagi && now.Minute()==
ldrMenitNyalaPagi ){
        ldrIndikator=1;
    }
    if (now.Hour() == ldrJamMatiMalam &&
now.Minute()==ldrMenitMatiMalam)
    {
        ldrIndikator=0;
    }
    if(ldrIndikator == 1)
    {
        growlight = 1;
    }
    if(ldrIndikator == 0){
        growlight = 0;
    }
}

void growlightAktif(){
    RtcDateTime now = RTC.GetDateTime();

```

```

if (growlight ==1){
    digitalWrite(relaypin, HIGH);
    bacaLdr = analogRead(LdrInputPin);
    if(bacaLdr >=10 ){
        bacaLdrKalibrasi = bacaLdr;
        if( bacaLdrKalibrasi >= 255){
            bacaLdrKalibrasi = 255;
        }
        // if (ba
    }else{
        bacaLdrKalibrasi =LOW;
        if(bacaLdr == 0){
            bacaLdrKalibrasi =LOW;
        }
    }
    Serial.print("baca LDR: ");
    Serial.println(bacaLdrKalibrasi);
    analogWrite(ldrOutputPin , bacaLdrKalibrasi );
}
if(growlight ==0){
    digitalWrite(relaypin, LOW);
    bacaLdrKalibrasi = LOW;
    analogWrite(ldrOutputPin , bacaLdrKalibrasi );
}
}
float konstanta;

void ECfinal(){
    konstanta = 1442.334;
    adcEC = analogRead(sensorECpin);
    Serial.print(" "); Serial.println(adcEC);
    x2=adcEC*adcEC;
    x3=adcEC*x2;
    h_x= 1.488*adcEC;
    h_x2=0.000898*x2;

```



```

h_x3= 0.000000184*x3;
nilaiEC1=konstanta-h_x;
nilaiEC2=h_x2-h_x3;
nilaiEC=nilaiEC1+nilaiEC2;
EC1= int(nilaiEC);
EC = String(EC1);
}

```

A.2. Program Android Studio

```

package com.tommyputranto.ec;

```

```

import android.os.Bundle;
import android.os.Handler;
import android.support.v4.app.Fragment;
import android.support.v4.widget.SwipeRefreshLayout;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

```

```

import com.github.mikephil.charting.charts.LineChart;
import com.github.mikephil.charting.components.LimitLine;
import com.github.mikephil.charting.components.YAxis;
import com.github.mikephil.charting.data.Entry;
import com.github.mikephil.charting.data.LineData;
import com.github.mikephil.charting.data.LineDataSet;
import com.github.mikephil.charting.utils.ColorTemplate;
import com.tommyputranto.ec.api.Api;
import com.tommyputranto.ec.model.Data;

```

```

import java.util.ArrayList;

import retrofit.Callback;
import retrofit.RestAdapter;
import retrofit.RetrofitError;
import retrofit.client.Response;

```

```

public class MainFragment extends Fragment {
    ArrayList<Entry> entries;
    private SwipeRefreshLayout mSwipeRefresh;

```

```

ArrayList<String> labels;
String[] label;
String API = "http://androbos.com";
private LineChart lineChart;
public static MainFragment newInstance() {
    return new MainFragment();
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setRetainInstance(true);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    View v = inflater.inflate(R.layout.main_fragment, container,
false);
    lineChart = (LineChart)v.findViewById(R.id.chart);
    mSwipeRefresh
(SwipeRefreshLayout)v.findViewById(R.id.swipe);
    mSwipeRefresh.setOnRefreshListener(new
SwipeRefreshLayout.OnRefreshListener() {
        @Override
        public void onRefresh() {
            refreshContent();
        }
    });
    loadJson();

    return v;
}
private void refreshContent(){
    new Handler().postDelayed(new Runnable() {
        @Override
        public void run() {

```

```

loadJson();
mSwipeRefresh.setRefreshing(false);
    }
    },3000);
}
private void loadJson() {
    RestAdapter restAdapter = new
    RestAdapter.Builder().setLogLevel(RestAdapter.LogLevel.FULL)
    .setEndpoint(API).build();
    Api api = restAdapter.create(Api.class);
    api.getFeed(new Callback<Data>() {
        @Override
        public void success(Data data, Response response) {
            entries = new ArrayList<>();
            labels = new ArrayList<String>();
            label = new String[]{
                "PPM"
            };
            for (int i = 0; i< data.getData().size();i++){
                //entries.add(new Entry(
                Float.parseFloat(data.getData().get(i).getHari()), i));
                entries.add(new Entry(
                Float.parseFloat(data.getData().get(i).getDataEC()), i));
                labels.add(data.getData().get(i).getJam());
            }
            LineDataSet dataset = new LineDataSet(entries, "# of
Calls");
            LineData datagraph = new LineData(labels, dataset);

            dataset.setColors(ColorTemplate.COLORFUL_COLORS); //
            dataset.setDrawCubic(true);
            dataset.setDrawFilled(true);

            lineChart.setData(datagraph);
            lineChart.animateY(5000);

            lineChart.setDescription("Perkembangan Nilai EC");

```

```

LimitLine line = new LimitLine(600);
LimitLine lineBawah = new LimitLine(400);
YAxis leftAxis = lineChart.getAxisLeft();

leftAxis.addLimitLine(line);
leftAxis.addLimitLine(lineBawah);

}

@Override
public void failure(RetrofitError error) {

}

});

}
}

```

A.3. Program Update Server

```

<?php
mysql_connect("localhost", "u2067583_sim900", "sim900") or
die(mysql_error());
mysql_select_db("u2067583_sim900") or die(mysql_error());

if($_GET["hari"]||
$_GET["jam"]||$_GET["tanggal"]||$_GET["dataEC"]||$_GET["dataLD
R"]||$_GET["dataPompa"]) {
$insert = "INSERT INTO android (hari, jam,
tanggal, dataEC, dataLDR, dataPompa) VALUES
('".$_GET['hari']."', '".$_GET['jam']."',
".$_GET['tanggal']."', '".$_GET['dataEC']."',
".$_GET['dataLDR']."',
".$_GET['dataPompa']."'");
if($insert){
echo "sukses ". $_GET['hari']." ".
$_GET['jam']." ". $_GET['tanggal']."
".$_GET['dataEC']." ".$_GET['dataLDR']."
".$_GET['dataPompa']."' ". "<br />";
}
}

```



```

    }
    $add_member = mysql_query($insert);
}
?>

```

A.4. Program *JSON*

```
<?php
```

```

mysql_connect("localhost", "u2067583_sim900", "sim900") or
die(mysql_error());
mysql_select_db("u2067583_sim900") or die(mysql_error());

    /// http://androbos.com/sim900/sim900/fix.php?data=8
    $sql = "SELECT id, hari, jam,tanggal,dataEC,
dataLDR,dataPompa FROM android ORDER BY id DESC LIMIT 50";
    $result = mysql_query($sql) or die ("Query error: " .
mysql_error());
    //fetch dalam bentuk array
    //$records = array();
    //while($row = mysql_fetch_assoc($result)) {
    // $records[] = $row;
    //}
    //menuliskannya dalam bentuk json menggunakan fungsi
    json_encode
    //echo $_GET['jsoncallback'] . '(' . json_encode($records) . ');';
    //echo $_GET['jsoncallback'] . json_encode($records) ;

    $arr = array();
    while ($row = mysql_fetch_assoc($result)) {
        $temp = array(
            "hari" => $row["hari"],
            "jam" => $row["jam"],
            "tanggal" => $row["tanggal"],

```

```
"dataEC" => $row["dataEC"],  
"dataLDR" => $row["dataLDR"],  
"dataPompa" => $row["dataPompa"], );  
    array_push($arr, $temp);  
}  
$arr1=array_reverse($arr);  
$data = json_encode($arr1);  
echo "{\data\":" . $data . " }";  
?>
```

DAFTAR RIWAYAT HIDUP



Nama : Tommy Dwi Putranto
TTL : Bandung, 31 Desember 1994
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Mustang No. 2 Sby
Telp/HP : 085334660023
E-mail : tommyputranto1@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SDN KPAD 1 Bandung
2. 2007 – 2010 : SMP GIKI 1 Surabaya
3. 2010 – 2013 : SMA Negeri 2 Surabaya
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT Srikandi Agung Perkasa

PENGALAMAN ORGANISASI

1. Google Developer Group
2. Google Student Group
3. Surabaya Youth



-----Halaman ini sengaja dikosongkan-----

DAFTAR RIWAYAT HIDUP



Nama : Bayu Fatchur Rohman
TTL : Madiun, 14 Februari 1995
Jenis Kelamin : Laki - laki
Agama : Islam
Alamat : Pucanganom RT. 35/03
Kebonsari, Madiun
Telp/HP : 085785766756
E-mail : bayurohman95@gmail.com

RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SDN Sidorejo 2
2. 2007 – 2010 : SMP Negeri 1 Dolopo
3. 2010 – 2013 : SMA Negeri 1 Geger
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di PT CNC Controller Indonesia, Bekasi

PENGALAMAN ORGANISASI

1. Sie Dokumentasi IARC 2014
2. Sie Dekorasi IARC 2015
3. Koordinator Asisten Laboratorium Elektronika Terapan D3 Teknik Elektro FTI ITS periode 2015/2016

